Numerical Simulation of Cyclic Thermodynamic Processes

Andersen, Stig Kildegård

Publication date:
2006

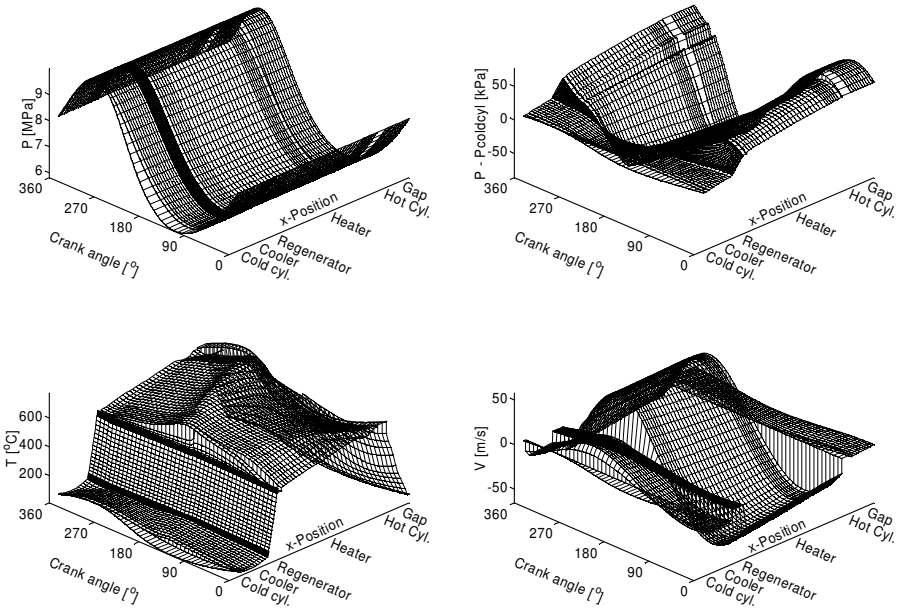
Document Version
Publisher's PDF, also known as Version of record

Technical
University of
Denmark

Numerical Simulation of Cyclic Thermodynamic Processes

Stig Kildegård Andersen

Ph.D. thesis
March 2006



Department of
Mechanical
Engineering

MEK

Energy
Engineering
Section

Numerical Simulation of Cyclic Thermodynamic Processes

Stig Kildegård Andersen

Technical University of Denmark
Department of Mechanical Engineering
Energy Engineering Section
March 2006

Published in Denmark by the
Technical University of Denmark

Energy Engineering Section
Department of Mechanical Engineering
Technical University of Denmark
Nils Koppels Alle, Building 402
DK-2800 Kgs. Lyngby
Phone: +45 4525 4122
E-mail: bh@mek.dtu.dk
WWW: http://www.et.mek.dtu.dk/

Publication Reference Data
Andersen, S. K.
Numerical Simulation of Cyclic Thermodynamic Processes
Ph.D. Thesis
Technical University of Denmark, Dept. of Mech. Eng., Energy Eng. Section
March, 2005
Report. No. MEK-ET-PHD-2006-01
ISBN 87-7475-334-7
Keywords: Simulation, compressible one-dimensional flow, periodic steady state, shooting, Stirling

Preface

This thesis was prepared at the Energy Engineering Section of the Department of Mechanical Engineering at the Technical University of Denmark in partial fulfilment of the requirement for acquiring the Ph.D. degree in engineering. The Ph.D. project was financed by the Ph.D. programme for Mechanical and Civil Engineering at The Technical University of Denmark and lasted from the 1st of September 2002 to the 27th of March 2006. The Ph.D. project was supervised by:

Professor Henrik Carlsen, Energy Engineering Section, Department of Mechanical Engineering, Technical University of Denmark.

Professor Per Grove Thomsen, Informatics and Mathematical Modelling, Technical University of Denmark.

The thesis deals with the development of methods for simulation of cyclic thermodynamic processes such as those which occur in Stirling machines and pulse tube coolers. The thesis consists of a report on the methods developed during the project and a collection of four research papers, published elsewhere, on applications of the methods.

Lyngby, March 2006

Stig Kildegård Andersen

(This page intentionally left blank)

Acknowledgements

I would like to thank my supervisors for their guidance, interest, and support during this work and the previous projects I carried out under their supervision.

I also wish to thank (in no particular order):

- Professor Günther Thummes and Marc Dietrich for my interesting stay at the University of Giessen.
- Dr. Ingo Rühlich for inviting me to Heilbronn and for making my stay both very interesting and pleasant.
- Professor Bernd Thomas for help with his *PROSA* software and for the interesting email-discussions we have had during the last years.
- Professor Hans-Detlev Kühl for help on numerical diffusion and help on acquiring literature.
- Bernd Dammann for repeated help with the Sun compiler.
- Those who helped me but preferred to remain anonymous - You know who you are.
- My friends and colleagues at the Technical University of Denmark.

Finally, I wish to thank my family for their interest and patience, and for their help in debugging this text.

(This page intentionally left blank)

Summary

This thesis is on numerical simulation of cyclic thermodynamic processes. A modelling approach and a method for finding periodic steady state solutions are described. Examples of applications are given in the form of four research papers.

Stirling machines and pulse tube coolers are introduced and a brief overview of the current state of the art in methods for simulating such machines is presented. It was found that different simulation approaches, which model the machines with different levels of detail, currently coexist. Methods using many simplifications can be easy to use and can provide results quickly, but they are limited with respect to the phenomena that can be studied. More comprehensive methods can be used to study and optimise machines or components in more detail, but they usually require more time and computer resources. In this work the focus was on methods that are fast enough to be used for numerical optimisation of complete machine designs. The highest level of detail which appears to be feasible for this purpose is to model the gas flows in the machines as being primarily one-dimensional.

The theory and implementation of a control volume based approach for modelling oscillating, compressible flow in one space dimension is presented. The implementation produces models where all the equations, which are on a form that should be understandable to someone with a background in engineering thermodynamics, can be accessed and modified individually. The implementation was designed to make models flexible and easy to modify, and to make simulations fast.

A high level of accuracy was achieved for integrations of a model created using the modelling approach; the accuracy depended on the settings for the numerical solvers in a very predictable way. Selection of fast numerical algorithms and multi-threading accelerated simulations considerably. The discretisation scheme of the modelling approach was found to be convergent, and even relatively coarse discretisations produced useable results. Models created using the modelling approach produced results in good agreement with experimental data, and with simulation results from current state of the art software, for two Stirling machines and two pulse tube coolers.

Parallelised single and multiple shooting methods were studied and were found to be reliable for finding periodic steady state solutions. Multiple shooting methods had better parallel scalability but this advantage was almost neutralised by a significant overhead compared to single shooting. The overhead was due to transients at the beginnings of the sub intervals in the cycle. The severity of the overhead was specific to models which included the inertia of the gas in the momentum balance. Single shooting, where the fastest evolving variables such as velocities and pressures, where excluded from the shooting, was the fastest sequential method. Fixed point iteration was performed on the excluded variables during the shooting. The parallel scalability for batch jobs was improved with an implementation which uses the parallelism inherent in batch jobs to increase the scalability of the parallelised shooting methods.

The four research papers are self contained studies on: 1) the effects of regenerator matrix temperature oscillations on the performance of a Stirling engine, 2) optimal regenerator designs which takes into account the effects of the regenerator matrix temperature oscillations, 3) transverse asymmetry in the temperature profile of a regenerator in a pulse tube cooler, and 4) the appendix gap losses in a Stirling engine.

(This page intentionally left blank)

Resume (Summary in Danish)

Denne afhandling omhandler numerisk simulering af cykliske termodynamiske processer. En modelformulering og en metode til at finde periodestationære løsninger er beskrevet. Eksempler på anvendelser er vist i fire videnskabelige artikler.

Der gives en introduktion til Stirlingmaskiner og pulse tube-kølere og en kort oversigt over status inden for metoder til simulering af sådanne maskiner. Det blev fundet, at der er flere forskellige aktuelle simuleringsmetoder, der modellerer maskinerne med forskellige detaljeringsgrader. Metoder med mange forsimplende antagelser kan være meget brugervenlige og kan give resultater hurtigt, men kan kun studere overordnede fænomener. Mere omfattende metoder kan bruges til nærmere analysere og optimering af maskiner og komponenter, men kræver oftest mere tid og regnekraft. I dette arbejde er der fokuseret på metoder, som er hurtige nok til at være anvendelige til numerisk optimering af komplette maskindesigns. Den fineste detaljeringsgrad, som er operativ i denne sammenhæng, ser ud til at være modellering af gasstrømningerne i maskinerne som værende hovedsageligt endimensionale.

Teorien bag og en implementering af en kontrolvolumenbaseret metode til modellering af endimensionale, oscillerende, kompressible strømninger præsenteres. Implementeringen giver modeller, hvor alle ligningerne, der er formuleret så de skulle være tilgængelige for personer med baggrund i anvendt termodynamik, kan tilgås og ændres enkeltvis. Implementeringen blev designet til at gøre modeller fleksible og lette at ændre, og samtidigt gøre simuleringer hurtige.

Stor nøjagtighed blev opnået ved numerisk integration af en model formuleret efter modelleringsmetoden. Sammenhængen mellem løserindstillinger og opnået nøjagtighed var meget forudsigelig. Udvælgelse af passende numeriske algoritmer og parallelisering vha. tråde forøgede simuleringshastigheden betydeligt. Det blev fundet at diskretiseringen i modelleringsmetoden var konvergent, og at selv relativt grove diskretiseringer gav brugbare resultater. Modeller lavet med modelleringsmetoden gav resultater, der stemte godt overens med eksperimentelle data og beregningsresultater fra state-of-the-art software, for to Stirlingmaskiner og to pulse tube kølere.

Paralleliserede enkelt- og multiple skydemetoder blev studeret, og blev fundet at være pålidelige til at finde periodestationære løsninger. Det blev fundet at multiple skydemetoder skalerede bedst på parallelcomputere, men at denne fordel stort set blev neutraliseret af at metoderne var mere regnetunge end enkeltskydemetoder. Den ekstra regnetid skyldtes transienter ved begyndelserne af underintervallerne i cyklussen. Disse transienter var særligt beregningstunge for modeller, der inkluderer gassens inerti i impulsbalancen. Den hurtigste metode ved sekventielle beregninger var enkelt skydning, hvor de hurtigste variable, såsom hastigheder og tryk, var ekskluderet fra skydningen. Der blev foretaget fikspunktiteration på de ekskluderede variable. Skalerbarheden for gruppekørsler blev forbedret med en implementering, som udnytter de indbyggede muligheder for parallelitet ved gruppekørsler til at forbedre skalerbarheden af de paralleliserede skydemetoder.

De fire videnskabelige artikler er selvstændige studier af: 1) Indflydelsen af regeneratormatrixtemperatursvingninger på en Stirlingmotors ydelse, 2) optimale regeneratormatrixdesigns som tager højde for regeneratormatrixtemperatursvingninger, 3) asymmetri på tværs af strømningsretningen i temperaturprofilet i en regenerator i en pulse tube køler, og 4) ringspalte-tabene i en Stirlingmotor.

(This page intentionally left blank)

Table of contents

Nomenclature

$A_{c,i}$	Cross section at centre of vol. *i* [m^2]
$A_{c,ref,i}$	Reference cross section in vol. *i* [m^2]
$A_{ht,i,k}$	Heat transfer area of surface segment *k* in vol. *i* [m^2]
c_v	Specific heat at constant volume [$J/(kg \cdot K)$]
$\underline{c}$	Parameters
$\underline{c_i}$	Parameters corresponding to integral conditions
$\Delta p_{f,i}$	Pressure drop due to friction in vol. *i* [Pa]
Δt	Cycle period [s]
Δt_{step}	Step size [s]
e_i	Specific total energy at centre of vol. *i* [J/kg]
E_i	Total energy in vol. *i* [J]
$F_{AD,i}$	Artificial dissipation force in vol. *i* [N]
$F_{wall,i}$	Wall friction force in vol. *i* [N]
$h_{conv,i,k}$	Average convective heat transfer coefficient between gas and surface segment *k* in vol. *i* [$W/\left(m^2 \cdot K\right)$]
$h_{gas,i}$	Enthalpy at centre of vol. *i* [J/kg]
$\underline{\underline{I}}$	Identity matrix
$\underline{\underline{IM}}$	Iteration matrix
$\underline{\underline{J}}$	Jacobian matrix
$\underline{\underline{J_{ms}}}$	Jacobian matrix for the multiple shooting method
$\underline{\underline{J_s}}$	Jacobian matrix for the single shooting method
$k_{eff,j}$	Effective heat conductivity, including turbulent mixing, at centre of volume *j* [$W/(m \cdot K)$]
Ψ_j	Flux limiter function at centre of volume *j* [–]
l_i	Length of vol. *i* [m]
λ	Scaling factor used in shooting methods
m_i	Mass in vol. *i* [kg]
$\dot{m}_i$	Mass flow rate at centre of vol. *i* relative to volume *i* [kg/s]
$\tilde{\mu}_1$	Artificial dissipation coefficient [s/m]
$\tilde{\mu}_2$	Artificial dissipation coefficient [s/m]
N_{dyn}	Number of dynamic variables
N_i	Number of integral conditions
N_p	Number of periodic dynamic variables
p_i	Pressure at centre of vol. *i* [Pa]
p_{avrg}	Average instantaneous pressure in computational domain [Pa]
$\tilde{p}_i$	Interpolated pressure at flow area change in vol. *i* [Pa]
p_{mean}	Cyclic mean pressure in one specific control volume [Pa]
r	Sensor variable using consecutive gradients
$\underline{r_s}$	Residuals for single shooting equation system
$\underline{r_{ms}}$	Residuals for multiple shooting equation system
R	Gas constant [$J/(kg \cdot K)$]
ρ_i	Gas density at centre of vol. *i* [kg/m^3]
$\underline{s}$	States

$SCTV$	Scaling condition target value used when performing shooting
$\underline{SE}$	Shared evaluations
t	Time [s]
t_0	Time where a cycle is defined to begin [s]
T_i	Temperature at centre of volume *i* [K]
$T_{w,i,k}$	Wall temperature of surface segment *k* in vol. *i* [K]
u_i	Specific internal energy in vol. *i* [J / kg]
U_i	Internal energy in vol. *i* [J]
v_i	Specific vol. at centre of vol. *i* [m^3 / kg]
V_i	Size of vol. *i* [m^3]
$\bar{V}_i$	Velocity at centre of vol. *i* relative to volume *i* [m/s]
$\bar{V}_{CS}$	Velocity in the x-direction of coordinate system relative to inertial coordinate system [m / s]
x	Axial space coordinate [m]
$\underline{x}$	Static variables
$\underline{y}$	Dynamic variables
$\underline{y_i}$	Integration variables for integral conditions
$\underline{y_p}$	Periodic dynamic variables
$\underline{y_p^s}$	Periodic dynamic variables that scale with the mean pressure
$\underline{y_p^{ns}}$	Periodic dynamic variables that do not scale with the mean pressure
$\underline{y_{p,0}}$	Initial values for periodic dynamic variables
y_s	Integration variable for scaling condition

(This page intentionally left blank)

1. Introduction

In this work we consider cyclic thermodynamic processes that will approach a periodic steady state, if they are left to run for a sufficiently large number of cycles. Such cyclic, or periodic, processes are abundant in engines, cooling machines, heat pumps, compressors, rotary devices, and so on.

A common feature of the above mentioned machines and devices is that their designs are often optimised for good performance at periodic steady state operation. If such a design optimisation is to be assisted by using a numerical simulation model then periodic steady state solutions to the numerical model are needed.

This can be problematic because the cyclic thermodynamic processes in the machines typically approach periodic steady state in an asymptotic manner. An engine, for instance, may need several minutes of running time after a cold start until steady state conditions are reached. To find a periodic steady state solution to a numerical model of such a machine it can sometimes be necessary to simulate many consecutive cycles of operation in order for the solution to evolve until it is close to the periodic steady state. Finding periodic steady state solutions to complex numerical models in this way may require many hours of CPU time. Waiting for extended periods of time when using simulation as a design aid is impractical.

One approach to reducing the waiting time is to relax the requirements for how close to periodic steady state that solutions need to be. But this approach may compromise the consistency of the solutions, i.e. it can become impossible to tell if changes in a solution are actually due to changed input parameters or if the changes are just noise due to deviations from periodic steady state. Good consistency is usually critical when using simulation results as input for a numerical optimisation routine.

A different approach to reducing the waiting time is to reduce the complexity of the numerical model. But if too many simplifying assumptions are made, then the ability of the model to mimic the actual machine may be compromised and that can reduce the usefulness of the model.

Another way to reduce the waiting time is to use specialised simulation software, which uses optimised numerical methods and/or parallel computing techniques to reduce the time needed to find periodic steady state solutions.

In the present work efforts have been divided between:

1. Developing a good way to make models that can predict the performance of machines, for which oscillating compressible flow that is predominantly one-dimensional plays an important role. Stirling machines have been used as the primary example, but pulse tube coolers have also been modelled. It was a priority that the models should be in a format which, when fully developed, would be flexible and yet simple enough to allow individual equations to be understood and modified by a user with a background in engineering thermodynamics.
2. Optimising a numerical method, namely the shooting method, for finding periodic steady state solutions to the models. It was a priority to be able to compute periodic steady state solutions quickly enough that numerical optimisation of complete machine designs were practical.
3. Applying the developed methods for performing numerical studies on phenomena and loss mechanisms in Stirling machines and pulse tube coolers.

1.1. The type of cyclic processes considered here

In this work we consider cyclic thermodynamic processes that will approach a periodic steady state if they are left to run for a sufficient number of cycles.

In such a process it would be possible to measure the value of some quantity, either at a fixed point in space or inside an imaginary fluid parcel, such that the measured value would never reach a constant value. Even when a zillion cycles had passed the measured value would still oscillate with a constant period. If one continuously plotted the measured value against the relative position in each cycle then the plotted curves for different cycles would be right on top of each other. If one plotted the measured value against time, starting at the time when the first cycle began, the resulting plot could look like Fig. 1. After an initial transient the cycles begin to look identical.

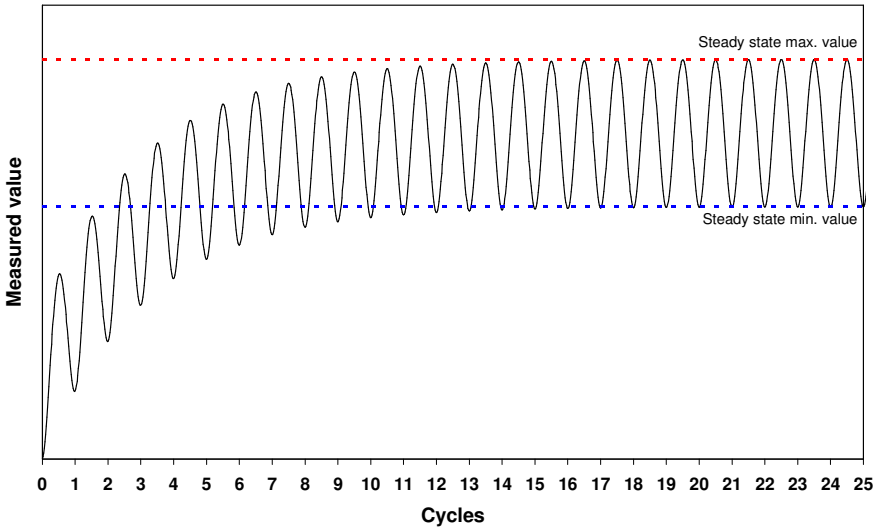


Fig. 1. Illustration of a measured value approaching a periodic steady state.

In the remainder of this thesis the terms “cyclic” and “periodic” are used to denote processes that behave as described above.

If the before mentioned measurement was made on a very small scale, say on the scale of the turbulence in the gas flows inside an engine, then the measured signal might have a chaotic component, so that there would never be two identical cycles. In such cases it is not trivial to formulate conditions for when periodic steady state has been reached. This problem is not investigated here.

1.2. Examples of machinery that embodies cyclic processes

In this work the processes that occur in Stirling engines, Stirling coolers, and pulse tube coolers have been studied as examples of cyclic thermodynamic processes. Stirling machines have been used as the primary example, but simulations of pulse tube coolers have also been performed. As an introduction to the material presented in the chapters that follow, Stirling machines and pulse tube coolers are briefly described below.

The theoretical stepwise thermodynamic cycles, which are used widely in textbooks on thermodynamics, have little resemblance to the processes that takes place inside actual

machines of the types described below. Therefore they are not discussed here. Instead the descriptions below are aimed at giving the reader an initial feel for the nature of the actual cyclic thermodynamic processes, that takes place inside the machines, and a feel for what is needed in order to successfully model the machines.

1.2.1. Stirling engines and Stirling coolers

Stirling engines and Stirling coolers are described here under the same heading because the working principles of the machines are very similar. For some production Stirling machines it is primarily the printed labels on the exterior of the machines and the materials chosen for their construction that enables one to determine if a Stirling machine is actually an engine or a cooler.

Stirling machines basically embody a serial connection of the following five components:

1. A compression volume.
2. A heat exchanger for heat rejection, the heat rejector.
3. A regenerator.
4. A heat exchanger for heat absorption, the heat absorber.
5. An expansion volume.

The serial connection of the components is illustrated in Fig. 2. The total volume contained inside the five components is denoted the working volume and it is filled with gas.

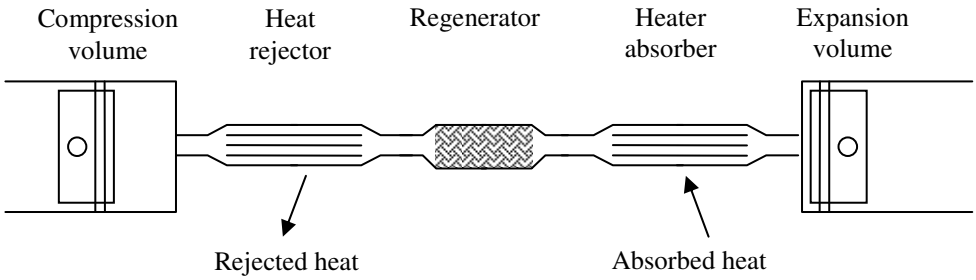


Fig. 2. The basic components in a Stirling machine.

Stirling machines as engines

In order for a Stirling machine to work as an engine the temperature of the heat absorber must be higher than the temperature of the heat rejector. The work producing cyclic process that takes place inside a Stirling engine can be mentally divided into four phases:

Phase 1: Work is expended to compress the gas while it is mainly in the compression volume.

Phase 2: The gas is moved from the compression volume to the expansion volume by moving both pistons simultaneously. During the transport through the heat exchangers and the regenerator the gas is heated so that it arrives in the expansion volume with a temperature near that of the hot heat absorber. This heating takes place at near constant volume and causes the pressure to rise.

Phase 3: The hot gas is expanded in the expansion volume. More work, than was needed for compression, can be extracted from the expansion because the pressure is higher.

Phase 4: The gas is moved back from the expansion volume to the compression volume. On its way the gas is cooled so that it has a temperature near that of the heat rejector when it arrives in the compression volume. The cooling occurs at near constant volume and causes the pressure to decrease. The cycle can now be repeated by beginning with phase 1.

One could make the four phases very distinct from each other if the pistons in Fig. 2 were moved in a step like manner. In a real machine, however, the phases change fluently from one to the next because the motions of the pistons are more sinusoidal than stepwise. But the basic idea of the process remains the same: Compress a cold gas, heat it to increase the pressure, expand it and extract more work than was needed for compression, cool the gas to decrease the pressure, and start the next cycle. One can say that a duty is paid in the form of heat at a high temperature in order to produce work.

Stirling machines as coolers

When a Stirling machine is used as a cooler the temperature of the heat absorber is lower than the temperature of the heat rejector. This is just the opposite of what was the case for the Stirling engine. The work consuming cyclic process that takes place inside a Stirling cooler can be mentally divided into the following four phases:

Phase 1: Work is expended to compress the gas while it is mainly in the compression volume.

Phase 2: The gas is moved from the compression volume to the expansion volume. During the transport through the heat exchangers and the regenerator the gas is cooled so that it arrives in the expansion volume with a temperature near that of the heat absorber. This cooling takes place at near constant volume and causes the pressure to decrease.

Phase 3: The cooled gas is expanded in the expansion volume. The expansion causes a further decrease in temperature. Less work, than was needed for compression, can be extracted from the expansion because the pressure is lower.

Phase 4: The cold gas is moved back from the expansion volume to the compression volume. On its way the gas is heated so that it has a temperature near that of the heat rejector when it arrives in the compression volume. The heating occurs at near constant volume and causes the pressure to increase. The cycle can now be repeated by beginning with phase 1.

In a Stirling cooler the idea is to: Compress a warm gas, reject the heat from the compression in the heat rejector, expand the gas to further decrease the temperature, let the cold gas absorb heat in the heat absorber, start the next cycle. The patterns of motion for the pistons are the same in a Stirling cooler as in a Stirling engine. But because the expansion occurs at a lower temperature than the compression the result of the process is different: A duty is paid in the form of work in order to move heat from a cold place to a warm place.

The regenerator

In both Stirling engines and Stirling coolers the regenerator plays a crucial role in the cycle. The regenerator typically contains a porous matrix, the regenerator matrix, which has a very large surface area in contact with the gas. Regenerator matrices can be made in many ways from, for instance, very thin wires, fibres, layers of dimpled or etched foil where gas can flow between and/or through the layers, or beds of small packed spheres. The function of the regenerator is to act as a thermal heat storage. The regenerator matrix must absorb heat when hot gas is blown through it in one direction and it must release the energy again when the flow direction is reversed and cold gas is blown through it in the other direction. The alternating blasts of hot and cold gas from different directions cause a steep temperature gradient to build up inside the regenerator during the cycle.

In a Stirling engine the regenerator absorbs heat from the gas when the gas is moved from the expansion volume to the compression volume. This helps to cool the gas so that less heat is rejected in the heat rejector. When the gas flow is reversed and the high pressure gas is blown towards the expansion volume the regenerator releases energy to the gas. This helps to heat the gas and reduces the amount of heat that must be put into the gas by the heat absorber in order to obtain the desired high temperature and pressure. The regenerator increases the efficiency of the engine because it reduces the amount of high temperature heat input to the heat absorber needed to produce a certain amount of work. The axial temperature distribution in a Stirling engine, including the regenerator, is illustrated schematically in Fig. 3.

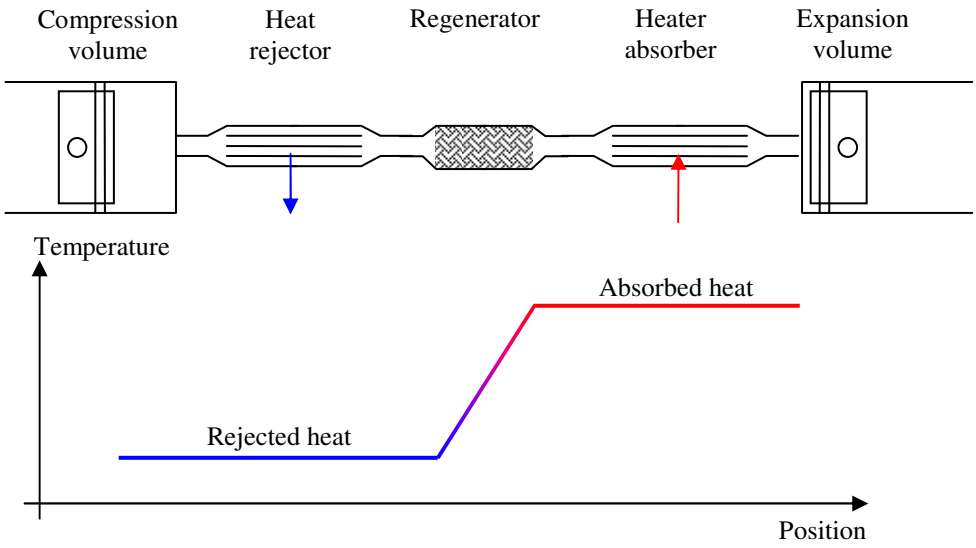


Fig. 3. Schematic of the axial temperature distribution in a Stirling engine.

In a Stirling cooler the regenerator absorbs heat when the compressed gas is moved from the compression volume to the expansion volume. This reduces the amount of heat carried by the gas from the warm end of the cooler to the cold heat absorber. The heat is released to the expanded gas again when the gas is pushed towards the heat rejector. The regenerator increases the coefficient of performance, i.e. the COP, because it increases the cooling power for a given amount of work input. The axial temperature distribution in a Stirling cooler is illustrated in Fig. 4.

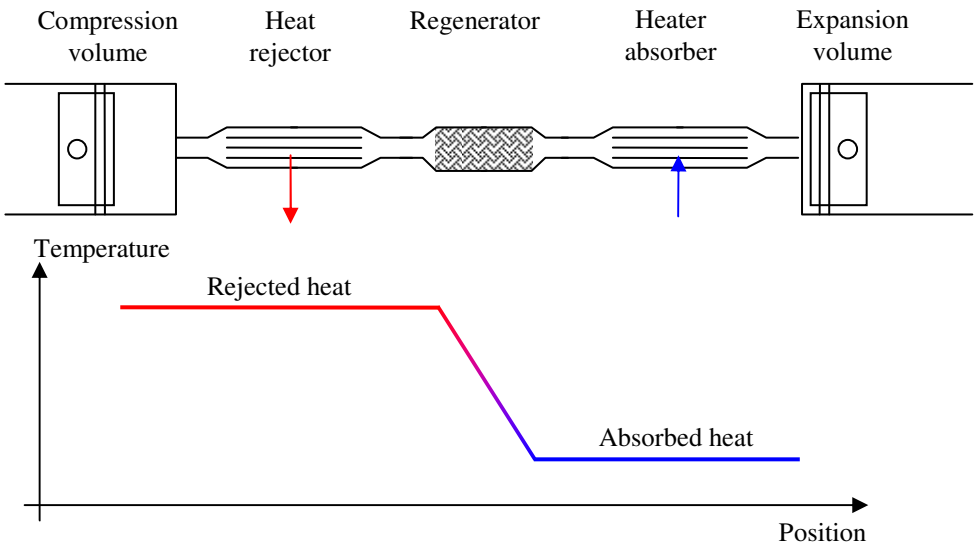


Fig. 4. Schematic of the axial temperature distribution in a Stirling cooler.

Real Stirling machines

In real Stirling machines the five basic components of a Stirling machine can be realised with different constructions, some of which look very different from the schematics above. There are also different principles for making the pistons move. But the basic working principles of the machines remain the same as has been outlined above.

1.2.2. Stirling type pulse tube coolers

Pulse tube coolers are typically used for cooling at cryogenic temperatures. There are different types and configurations of pulse tube coolers and this introduction is limited to the so called Stirling type pulse tube cooler with inertance tube. Such a Stirling type pulse tube cooler can consist of a serial connection of the following components:

1. A pressure wave generator.
2. A hot heat exchanger for heat rejection, called the *aftercooler*. The aftercooler removes most of the energy put into the gas by the pressure wave generator.
3. A regenerator.
4. A cold heat exchanger for heat absorption.
5. A pulse tube.
6. A hot heat exchanger for heat rejection. This heat exchanger primarily rejects the heat absorbed in the cold heat exchanger.
7. An inertance tube.
8. A buffer volume.

The components are connected in a serial connection as illustrated in Fig. 5, which also shows a schematic of the axial temperature distribution in a pulse tube cooler.

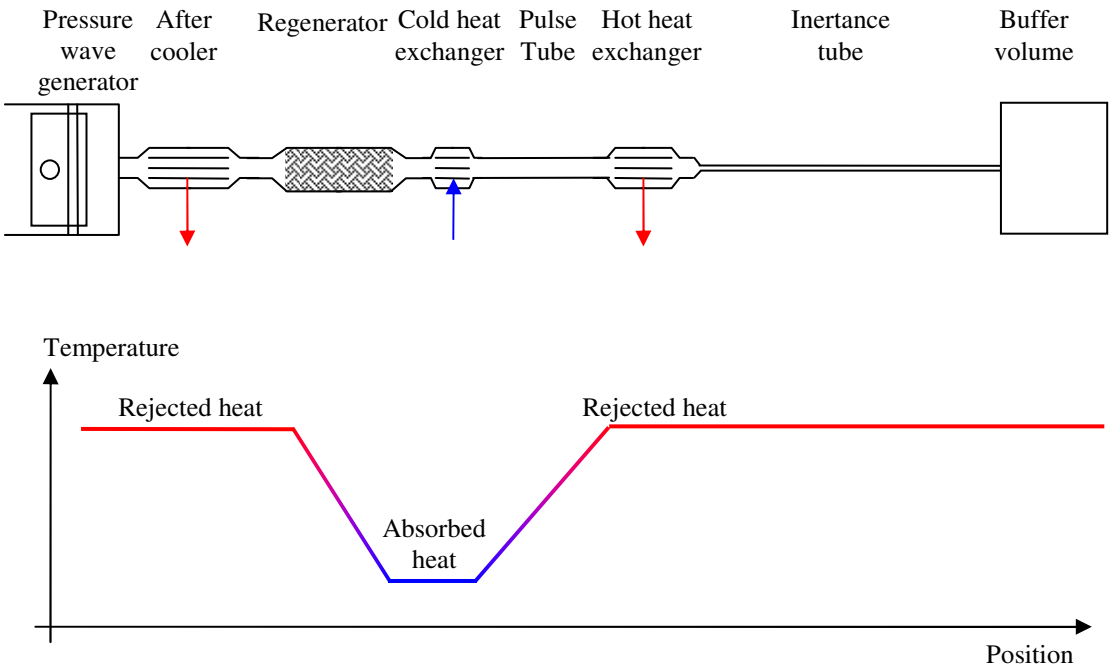


Fig. 5. The basic components in a Stirling type pulse tube cooler and a schematic of the axial temperature distribution.

When describing Stirling machines it was possible to mentally divide the processes in the machines into four phases: Compression, flow towards the expansion end, expansion, and flow towards the compression end. In Stirling machines the separation, or phase angle, between the pressure oscillation and the mass flow oscillation was controlled by changing the sizes of the cylinder volumes out of phase.

In a pulse tube cooler there is only one volume that varies in size, i.e. the volume in the pressure wave generator, and it is the pulse tube that provides a way for the oscillating gas flow to cool the cold heat exchanger. A pulse tube cooler has no moving parts at the cold temperature and this can improve the reliability when operating at cryogenic temperatures. It is also possible to have very low levels of vibration in the cold heat exchanger.

If a tube, with a moderate length to diameter ratio, is sealed off at one end and is open to an oscillating pressure wave at the other end, then the amount of gas inside the tube will oscillate almost in phase with the pressure; the mass flow oscillation at any axial position in the tube is hence approximately 90 degrees out of phase with the pressure oscillation. In this situation only a small amount of energy is transported along the tube towards the closed end, and the energy transport is due to interaction between the gas and the wall (Bauwens, 1996).

If a small opening is made at the closed end of the duct then this opening will add a mass flow oscillation that is in phase with the pressure oscillation. Gas will flow towards the small opening while it is at high pressure and temperature, and gas will flow towards the open end of the tube when it is at low pressure and temperature. In this way a significant axial energy transport towards the small opening is induced in the tube. It is this effect that is used to generate the cooling in a pulse tube cooler of the type shown in Fig. 5.

The function of the aftercooler is to remove the heat of compression. The function of the regenerator is, as it is in a Stirling cooler, to reduce the net amount of heat transported

by the oscillating flow towards the cold heat exchanger. The buffer volume and the inertance tube serve to provide a mass flow that is almost in phase with the pressure oscillation, so that a significant amount of energy is transported along the pulse tube from the cold heat exchanger to the hot heat exchanger. The inertance tube can be several meters long even when all the components from the aftercooler to the hot heat exchanger are packed inside a canister the size of a mans finger.

1.3. Gas flows in Stirling machines and pulse tube coolers

In order to successfully model a Stirling machine it is necessary to describe with equations what happens with the gas in the working volume of the machine during operation. If one chooses a level of detail where the actual gas flows are modelled then one must be able to model oscillating, compressible flow in heat exchangers, in a porous matrix, and in cylinder volumes. One must also model the solids in the machine with sufficient detail to determine the surface temperatures inside the machine, because the surface temperatures are important to the heat transfer processes that occur inside the machine.

In order to successfully model a pulse tube cooler of the type described above one must describe with equations more or less the same phenomena as when modelling Stirling machines. But with pulse tube coolers it is more critical to take into account the inertia of the gas, because it can have an important influence on the phase angles between the pressure oscillation and the mass flow oscillation at different locations in the cooler.

Frequently the designs of Stirling machines and pulse tube coolers are such that the gas flows in many of the components of the machines are approximately one-dimensional. This makes it possible for models with a one-dimensional discretisation of the gas flows inside the machines to predict the performance of the machines with good accuracy.

One-dimensional models need empirical correlations for flow friction and heat transfer in order to describe the interaction between the gas and the surfaces inside the machines. Correlations can also be used to describe axial transport phenomena that are due to multi-dimensional flow patterns, such as enhanced axial energy transport due to turbulent mixing or energy transport due to circulating flows inside a pulse tube.

It can, however, be problematic to find correlations that properly take into account the effects of, for instance, special inlet conditions on the circulating flows inside a pulse tube, or special inflow conditions where gas enters cylinder volumes. Therefore one-dimensional models may sometimes predict machine performance with insufficient accuracy for some purposes. Also, one-dimensional models do not allow one to study the multi-dimensional phenomena in detail so that, for instance, the geometry of an inlet could be optimised. One-dimensional models can, at best, allow one to take the net effect of the multi-dimensional phenomena into account through the use of empirical correlations.

2. Background and state of the art in methods for Stirling machine simulation

2.1. Introduction to Stirling machine modelling

Computer simulation has been used for design and optimisation of Stirling machines for several decades. Early simulation models were of limited complexities, in order to limit the need for processing time and computer memory, so that the models could run on the computers of the day. As could have been expected the complexities of simulation models have increased as computers have become ever more powerful. At the present day some scientists simulate gas flows inside complete Stirling machines using commercially available 3-dimensional Computational Fluid Dynamics packages (hereafter abbreviated 3D CFD).

But some of the simpler and faster modelling approaches, that were feasible decades before 3D CFD could even be considered for Stirling engine modelling, have not been made obsolete by the advent of the newer and more complex methods. On the contrary, simulation programs based on the simpler and faster modelling approaches are still being developed and requested commercially. One reason for this is that simulating a single cycle of a Stirling machine using 3D CFD still requires many CPU hours. This makes 3D CFD a very cumbersome tool for a number of design and optimisation purposes, especially when periodic steady state solutions are needed.

The simpler modelling approaches can yield results in anywhere from a few CPU hours to mere milliseconds, depending on the exact approach used. Also, the amount of input data needed to simulate a machine tends to increase with the complexity of a model. Finally, it may, in practice, prove necessary to have some understanding of the numerical methods used in a more complex model in order to successfully use the model. The simpler and faster modelling approaches can hence be implemented so that they require comparatively little effort, knowledge, or patience from the user. This makes them attractive for a number of simulation purposes.

The simple and very fast modelling approaches can sometimes achieve good agreement with experimental data for real machines. There is no general guarantee that using a more complex and computationally intensive method will yield results that agree better with experimental data.

The more complex and computationally intensive modelling approaches do have some very compelling advantages, though. A complex model can allow one to study the effects of lower level phenomena, which are not even considered in the simpler modelling approaches. Such phenomena include the effects of temperature distributions or geometric details in the interior of a machine on various losses. Also, a complex model may require fewer assumptions and experience factors than the simpler modelling approaches. A good implementation of a complex model can be very good at performance prediction as long as it is not used for modelling a design where a phenomenon, which is not properly accounted for in the model, is important.

No single modelling approach for Stirling type machines appears superior, or even appropriate, for all common simulation tasks. This is probably the main reason behind the present diversity in the coexisting methods that are used for modelling Stirling machines at the present day.

2.2. The scope of this chapter

A large number of methods and programs exist for simulating Stirling type and related machines. A few of these programs, namely *SNAPpro*, *PROSA*, and *Sage*, which are all discussed below, are commercially available. Other simulation programs are in-house codes of universities and institutions (like NASA), or are the intellectual property of persons and companies working in Stirling related businesses. A number of the programs have been described in varying levels of detail in the published literature but, to my best knowledge, none of these programs are freely available.

It would be difficult, expensive, very laborious, and hence somewhat misplaced in the context of this work, to attempt to produce a complete overview and comparison of the existing and previous modelling approaches and programs. Martini (1978) included a 100 page review of the then existing Stirling modelling methods (with some examples) in his now famous *Stirling Engine Design Manual* and concluded that his review was incomplete and should be expanded. The number of modelling methods in use and the number of implementations of the methods have increased since 1978. Other reviews and comparisons of older simulation methods can be found, for instance, in the papers of Ash and Heames (1981) and Urieli (1983).

This chapter is not a complete overview. The chapter briefly mentions a few historical programs and some of the important and relatively recent Stirling simulation efforts known to me. The chapter may serve as a brief introduction to the subject. But its main purpose is to provide sufficient background information to enable us to classify this work relative to the present day state of the art in methods for simulating Stirling and related machines.

This chapter contains a section that establishes an operative classification scheme for simulation methods for Stirling type and related machines. This is followed by sections that summarise relatively recent Stirling simulation methods divided into the categories of the abovementioned classification scheme.

2.3. Classification of simulation methods

Martini (1978) introduced a classification scheme for Stirling engine models in his *Stirling Engine Design Manual*. In the classification scheme models are divided into first, second, and third order models in the following way:

- *First order models:* Such a model is either "... a theoretical stepwise engine model." or methods for "... calculating loss-free engine output when the engine is crank operated ...". The latter group contains analysis methods such as Schmidt analysis, which is discussed in the manual, where the cycle no longer consists of step wise thermodynamic processes. These modelling approaches are defined as first order models when combined with experience factors. First order models yield closed form solutions.
- *Second order models:* These models are based on Schmidt analysis or something similar, i.e. on a simplified analysis of an idealised thermodynamic cycle that yields closed form solutions. In second order models various parasitic power losses and heat losses are calculated independently of each other and are used to correct the results from the analysis of the idealised cycle. Since the losses are assumed independent of each other the second order models also yield closed form solutions. The closed form solutions are sometimes iterated in order to correct the temperatures of the underlying idealised thermodynamic cycle with

calculated temperature differences between the gas and the heat exchanger walls (Lista, 1993b).

- *Third order models:* In third order models the working volume of the engine is divided into a number of nodes and the basic differential equations that govern the engine, i.e. the equations for mass, energy, and momentum, are solved using numerical methods. Third order models make fewer assumptions about the engine cycle and about the coupling of loss mechanisms to each other and to the engine cycle. This means that third order models may yield more accurate prediction of engine performance. Third order models do not yield closed form solutions and are much more computationally intensive than first and second order models.

The *Stirling Engine Design Manual* was revised and expanded and was published in a second edition by Martini (1983). In the second edition the definitions of the different orders in the classification scheme were reformulated so that the definitions were also based on possible uses for models of different orders. Perhaps this reformulation is part of the reason why some people perceive the definitions of order as being somewhat fuzzy.

Present day Stirling simulation tools seem to fit mainly into the second order and third order categories of the classification scheme from the 1978 version of the *Stirling Engine Design Manual*. The definition of third order models now appears very broad as it encompasses models ranging from simple 1D models with many assumptions and simplifications all the way up to fully compressible 3D CFD.

The 1978 classification by Martini is widely used within the Stirling community in spite of its age, its inaccuracies, and the fact that it is possible to confuse the term “order” with the numerical order of the discretisation of space and time in a third order model. Though the classification was developed by Martini for analysis methods for Stirling engines it appears that the classification is now also used for models of Stirling coolers and even pulse tube machines. Applying the same terminology for models of Stirling engines and Stirling coolers makes sense because there is very little difference between modelling a Stirling engine and a Stirling cooler. Also, some of the tools developed for simulation of Stirling machines have been developed so that they can also be used for pulse tube machines.

It is my impression, that the term “third order model” is now used mostly for models where the governing equations for fluid flow are discretised only in one space dimension and time, i.e. for models that could be described as 1D CFD. Please note, though, that third order models do not need to be strictly one-dimensional even though they use a 1D discretisation of the governing equations for fluid flow. Third order models can have multiple parallel flow paths, and the heat conduction in the solids of a machine can also be modelled as multi-dimensional. When 2D CFD is used it is usually referred to as “2D CFD”. When 3D CFD is used it is referred to either as “3D CFD” or simply “CFD”.

The current operative classification scheme, which will also be used in this thesis, thus appears to contain the following categories:

- First order models: Closed form solutions without parasitic losses.
- Second order models: Closed form solutions with parasitic losses.
- Third order models: Any form of 1D discretisation of the governing equations for fluid flow (no closed form solution).
- 2D CFD
- 3D CFD

The following sections describe simulation efforts belonging to the last four of these categories.

2.4. Second order simulation programs

2.4.1. The Martini-Weiss program by Martini

The *Martini-Weiss* program (sometimes the name is shortened to *MarWeiss*) was a simulation program for Stirling engines developed at the University of Calgary by Martini and later expanded by Weiss (Walker et al., 1990). The program was based on the work of Martini (1978, 1983) on second order modelling. There was also a version of the program called *CryoWeiss* for simulating cryocoolers. Successful validations of the Martini-Weiss program against experimental data for real engines have been published (Lista, 1993b). The *Martini-Weiss* program was commercially available in the late 80's and part of the 90's, but it appears that it has been surpassed and outdated by newer second order simulation programs.

2.4.2. The SNAPpro software by Altman

The *SNAPpro* software (*SNAP* is an abbreviation for *Stirling Numerical Analysis Program*) is a commercially available Stirling simulation program by Altman (2005). The program is based on the work of Martini (1978, 1983) and is classified as a second order model. The program is being actively promoted on the internet (Altman, 2005) and at conferences (Altman, 2003).

SNAPpro is written in *Excel* and uses the plotting capabilities of *Excel* to provide extensive plotting facilities. The program has facilities for simulating user defined Stirling designs, for performing parameter studies, and for performing optimisation using a genetic algorithm.

2.4.3. The PROSA software by Thomas

PROSA is an abbreviation of *Program for second order analysis*. *PROSA* is a commercial Stirling simulation program available from Thomas (2006). The versions of *PROSA* up to version 2.4 are second order models. But a new version 3.0, which will be a third order model that simulates using time stepping instead of using closed form solutions, is currently being developed.

Descriptions of *PROSA*'s second order analysis method, graphical user interface, and additional tools (for parameter studies, optimisation, and automatic calibration against experimental data) have been published in papers by Thomas (2001, 2003).

PROSA has been successfully validated, both with and without calibration, against experimental data for a number of Stirling engines as documented in the above mentioned papers by Thomas and in the work of Prieto and Garcia (2005).

2.5. Third order (1D) simulation programs

2.5.1. The MS*2 code by Bauwens

References to the *MS*2* Stirling simulation code by Bauwens appear in the literature up until the mid nineties, see for instance the papers by Bauwens (1990, 1993a, 1993b) and by Mitchell and Bauwens (1990). The *MS*2* code was a third order model and was based on a finite difference discretisation in one space dimension. The model used an Eulerian grid, i.e. a grid with non-moving nodes. Periodic steady state solutions were found by marching the solution forward in time from a given set of initial values until the solution became periodic (Bauwens, 1990). The sound numerical properties of the *MS*2* code have been documented in several papers (Mitchell and Bauwens, 1990) (Bauwens 1993a, 1993b) and validations of the code against experimental data has also been published (Mitchell and Bauwens, 1990). In the early nineties powerful workstations or super computers were needed when refining the discretisation of time and space to explore and prove the convergence of the discretisation scheme used in the code.

2.5.2. The simulation program by Kühl

Kühl (1990) described the simulation program *kpsim* that could perform both second order and third order analysis of machines with regenerative cyclic gas processes. The program allowed models to be built from a library of predefined components. The components were discretised in one space dimension on Eulerian grids. The program performed time stepping with the models when performing simulations. A method, which was based on understanding of the physics of regenerators at periodic steady state, was used to accelerate the convergence of axial matrix temperature distributions towards periodic steady state. The performance prediction of the program was successfully validated against experimental data for a Vuilleumier heat pump and two Stirling engines. The program has also been successfully validated against high speed measurements of instantaneous gas temperatures inside a Vuilleumier heat pump (Kühl et al., 1997).

2.5.3. Stirling simulation related to NASA (HFAST, GLIMPS, and SAGE)

NASA has been involved in Stirling engine research since the early 70's (NASA Glenn Research Center, 2005) and has been a significant sponsor for the development of Stirling simulation programs. NASA´s involvement in 1D Stirling models has been summarised by Ibrahim et al. (2001).

NASA developed its own 1D finite difference model in the late 70's and early 80's. Later NASA acquired unlimited rights the simulation program *HFAST* from Mechanical Technology, Inc. (MTI) of Albany, New York. NASA also purchased the Stirling simulation code *GLIMPS* by Gedeon. *HFAST* and *GLIMPS* were both more

time efficient and user friendly than the finite difference code previously developed at NASA. *HFAST* and *GLIMPS* were, together with *MS*2* code by Bauwens, considered to be the state of the art in 1994 according to Commiso (1994). Several papers on experimental validations and comparison of the two codes can be found in the literature (Geng and Tew, 1992)(Lista, 1993a)(Cairelli, Geng and Skupinski, 1989) (Cairelli et al., 1990).

In 1994 Gedeon introduced a simulation program named *Sage*. *Sage* is a direct descendant of the *GLIMPS* code. According to Tew et al. (2004) NASA devoted an increased level of effort to simulating several Stirling engines and a thermoacoustic engine with the *Sage* simulation program. According to Ibrahim et al. (2001) *Sage* had also become the primary design tool of the Stirling Technology Co. (or STC, which has now changed its name to Infinia).

HFAST

HFAST was a descendant of the *Harmonic Stirling Cycle Analysis Code* (*HSCAC*). Development of *HSCAC* started in 1978 at MTI in order to satisfy the need for a fast and reasonably accurate tool for design and optimisation (Rauch, 1980). The principle idea behind the solution method used in *HSCAC* was to make sufficient simplifying assumptions about the governing equations for mass, energy, and momentum that they could be solved analytically for harmonic functions. Rauch (1980) made a detailed description of the primary simplifying assumptions in the *HSCAC* code along with its iterative solution method. The solution process began by making an initial guess for the temperature distribution. The solution method then iterated the following four stages.

1. Solving the continuity equation for the pressure wave and mass flow. Here it was assumed that the gas temperatures were constant in the regenerator and heat exchangers, and that compression and expansion was adiabatic in the cylinder spaces.
2. Solving the momentum equation for the pressure drop. The pressure drop was then used to correct the pressures in the volumes in the model.
3. Solving the energy equation for the gas temperatures. Here it was assumed that the working gas flowed back and forth through the heat exchangers in two steady blows.
4. Solving additional energy equations for the wall temperatures.

When the iterations had converged the resulting solution was corrected with parasitic losses for leakage past piston seals, thermal hysteresis in the expansion and compression spaces, and shuttle conduction along the displacer.

HSCAC was revised several times. The name was changed to *FAST** to indicate that the solution method was potentially much faster than some alternative solution methods that needed time stepping. Tests at MTI indicated that *FAST** needed to be upgraded so that the governing equations for mass, energy, and momentum were solved with fewer assumptions in order to become a useful performance prediction tool (Huang, 1992). Due to significant difficulties the upgrade was made in stages until *HFAST 2.0* appeared in the late 80's. In the new version a one-dimensional discretisation based on a staggered mesh of control volumes was used and the solution method had been completely revised. Solutions now included time varying temperatures, and they obeyed energy conservation for the gas flows. According to Huang (1992) the version of *HFAST* available in 1992 had exploited the full potential of the harmonic analysis method for Stirling cycle simulation and the performance prediction of the code had improved.

GLIMPS

GLIMPS was an abbreviation for *Globally Implicit Stirling Cycle Simulation*. Gedeon (1986) described the *GLIMPS* code which he had developed with the goal of making modelling of Stirling machines with a 1D discretisation of space feasible on personal computers.

GLIMPS used a 1D finite difference mesh in space for the governing equations for mass, energy, and momentum in the gas and for energy in the solids of a Stirling machine. The governing equations were also discretised in time, so that the complete discretisation produced a 2D finite difference mesh. For each point in time in the 2D finite difference mesh the mesh had nodes corresponding to all the nodes in the 1D spatial discretisation. Each node in the 2D finite difference mesh hence had both a time and a space coordinate. At each node in the 2D mesh there was a corresponding solution point for each of the governing equations for mass, energy, and momentum.

In *GLIMPS* two opposing sides of the 2D finite difference mesh were connected and an additional condition for the mean pressure was added. This produced a globally implicit equation system, which could be solved all at once for the periodic steady state solution. This eliminated the need to simulate many cycles with a time stepping procedure until a periodic steady state solution was reached.

The solution method used in *GLIMPS* was advantageous because the implicit equation system made it possible to use as few as 10 time nodes while still maintaining the stability of the numerical scheme. This kept the total number of equations in the global equation system, and hence the required computational resources to find solutions, on a tolerable level. The ability to use a small number of nodes made *GLIMPS* fast, but using only very few nodes was not good for the accuracy of the solutions. *GLIMPS* required only minor assumptions and simplifications for the governing equations, such as excluding axial conduction in the gas, in order to make the equation system solvable.

Sage

Sage was developed by Gedeon from *GLIMPS*. Gedeon (1994) described how the computational core in the mathematical model of the Stirling process was essentially that of the *GLIMPS* code with a few changes. But the graphical user interface and the way that the equation system was generated behind the user interface was new. Models were now constructed by dragging icons representing predefined models of components onto a canvas in the graphical user interface. Mass flows paths and heat conduction paths between the components were then established by mouse clicking in the user interface. The graphical user interface enabled users to make endless permutations of possible machine designs by adding or removing components and by connecting the components in different ways.

Sage is, in my opinion, the leading (and only) commercially available tool for third order simulation of Stirling, pulse tube, and similar machines. Because *Sage* is expensive commercial software there is limited publicly available information about it. I have had the privilege to get a little first hand experience with *Sage,* to read parts of the *Sage* documentation, and to talk and work with experienced *Sage* users. Because *Sage* represents the current state of the art in third order simulation programs a short summary of my experiences with *Sage* appears relevant. The information contained in the following two paragraphs cannot be referenced to published literature.

Sage has a very impressive user interface and users seem to need little experience to modify models and perform simulations. The graphical user interface of

Sage allows models of high complexity to be constructed with relative ease. The solution process is relatively robust but Sage is computationally very intensive if accurate solutions are required. The requirements for memory and, especially, processing time increase sharply when the total numbers of space nodes or time nodes are increased. The bulk of the processing time is spent doing sparse decomposition of the Jacobian matrix for the very large globally implicit equation system that results from having many time and space nodes in a model. Experiments to determine how many nodes are needed for the solution to become reasonably independent of the discretisation is something that is left to run overnight with different numbers of nodes on different PCs.

The finite difference method in Sage is formulated so that the nodes in the temporal discretisation eliminate harmonics of progressively higher order. Two nodes are required to add one harmonic, i.e. a cosine with both amplitude and phase angle at a frequency that is a multiple of the operating frequency of the machine. So to get, for example, the first 7 harmonics one needs 14 time nodes. Usually, the first few harmonics dominate when looking at the harmonic series for individual variables in the solutions. This means that one can quickly compute solutions containing only a handful of harmonics that capture important temporal aspects of the solution. But in practice it can be a very time consuming task to compute a solution with a discretisation so fine that aggregated values such as, say, the cooling power of a pulse tube cooler does not change significantly when the numbers of time and space nodes are increased.

2.5.4. Alternative methods for third order modelling

The third order models referenced above all use discretisations in space, where gas is transported by flow relative to the nodes or control volumes in the discretisation in space. These methods appear to be the most widely used but alternative approaches to solving the governing equations for mass, energy, and momentum do exist. To my best knowledge, no programs that use these alternative methods have been widely adopted. But because the methods are fundamentally different from the methods presented above and because they do appear in the recent literature they are briefly mentioned here as background information.

One alternative approach is called Lagrangian analysis. In this approach the gas nodes always contain the same mass of gas and the gas nodes move relative to the solid nodes. One can picture this approach as a string of balloons that slide back and forth through the components of the simulated machine. Martini (1986) reported on an implementation of a Lagrangian analysis program for Stirling engines, but concluded that his implementation was incomplete and not suited for use in design.

Another method called “the lambda-scheme method” was used by Rispoli (1985) to model a simplified device containing some of the components of a Stirling engine. Calandrelli and Rispoli (1995) reported that the work started by Rispoli had been upgraded to a full five-volume model of a Stirling engine. According to Organ (1992), who strongly advocates use of “the method of characteristics” for solving the governing equations for fluid flow in Stirling machines, the lambda-scheme employed by Rispoli is basically also a version of the method of characteristics.

2.6. 2D and 3D CFD modelling

2.6.1. CFD modelling at NASA (CAST, CFD-ACE+)

CFD modelling of Stirling machines appears to have begun in the late 80's with work on 2D CFD simulations of components in Stirling machines. Gedeon (1989) reported using a code named *MANIFEST* (short for Manifold and Estimate) on a Cray supercomputer to model gas jets impinging on Stirling regenerators. Ibrahim, Tew and Dudenhoefer (1989) reported beginning work on modelling Stirling heat exchangers with a 2D-code, and in 1990 they reported using their code to model components in NASA´s *Space Power Research Engine* (often abbreviated SPRE) (Ibrahim, Tew and Dudenhoefer, 1990). Ibrahim et al. (2001) described the evolution from 1989 up to 2001 of a 2D CFD code named *CAST* for modelling components in Stirling machines, and mentions simulation of Stirling machine components using the commercial CFD package *CFD-ACE+*. In the paper Ibrahim et al. compared results for gas spring hysteresis losses in a free piston machine to experimental results, as well as results for heat transfer pr. unit area at different locations in a heat exchanger under oscillating flow conditions. Ibrahim et al. (2001) also described how the *CAST* code was used for assisting the making of an empirical method for predicting the transition between laminar and turbulent flow in Stirling machines. This empirical method for predicting transition is now used in *Sage*.

NASA received a 2D CFD model of a complete Stirling engine in 2003 (Tew et al., 2004). The model was developed at Cleveland State University under a NASA grant using the *CFD-ACE+* package. In solutions, which had not reached periodic steady state, flow phenomena such as flow in different directions in parallel tubes in heat exchangers had been observed. Such a phenomena could be a significant loss mechanism, if it also existed in periodic steady state solutions and in the real engine. Such a phenomenon would not be revealed by a strictly 1D third order model without parallel flow paths.

Tew et al. (2004) reported on several problems in using the CFD Stirling model. The problems included the need to simulate many cycles in order to reach a periodic steady state solution for the regenerator temperature profile. In 2004 NASA was using a 32 processor cluster for performing CFD simulations of Stirling machines.

2.6.2. 3D CFD modelling in Fluent by Mahkamov

Mahkamov and Ingham (2000) reported on using a 2D CFD model of a complete Stirling engine using an in-house CFD code. In a solution to the 2D CFD model it was observed that gas temperatures varied significantly with both radial and axial position within the cylinder volumes of the studied 1 kW V-type Stirling engine for solar applications.

In 2003 Mahkamov reported on 3D CFD modelling of two different Stirling engines using the commercial CFD package *Fluent* (Mahkamov and Djumanov, 2003)(Mahkamov, Djumanov, and Hislop, 2003). In both papers it was emphasised that the results from the 3D CFD calculations could be used to identify flow resistances that lead to significant losses. In (Mahkamov, Djumanov, and Hislop, 2003), for instance, it was found that a piston could partially block the flow through a connecting tube from a

cylinder volume, because the passage that was intended to let gas flow around the piston was too small. Chow and Mahkamov (2005) documented that they had successfully completed a feasibility study on a 3D CFD simulation of a coupled system with a burner and a Stirling engine. They had successfully simulated 64 engine cycles and obtained an only slowly changing solution. The results showed that the temperature distribution on the heater heads of the Stirling engine in the combustion chamber was non-uniform.

2.7. Summary of the present state of the art

The simulation tools currently in use can be divided into second order models, third order models, and multi-dimensional CFD models.

Second order models use closed form solutions for an idealised thermodynamic cycle and correct the results for a number of independent parasitic power- and heat losses. Commercially available second order analysis programs, such as *SNAPpro* and *PROSA*, are still being developed and used. Such programs are very fast, robust, and convenient to use for some simulation purposes. The programs use empirically determined correlations, which have been refined for three decades, for computing the parasitic losses.

The programs can sometimes produce results that agree well with experimental data. The agreement with experimental data can be particularly good within a narrow parameter space if models can be calibrated against experimental data. But the nature of the second order models limits their usefulness for studying lower level phenomena and for studying machines where different loss mechanisms are strongly coupled.

Third order models use a discretisation of the governing equations for fluid flow in one space dimension and time. Third order models require fewer assumptions than second order models but they must also rely on the validity of empirical correlations for computing flow friction and heat transfer.

The performance prediction of third order models can be superior to second order models if the third order models account for all the important phenomena in the simulated machine. Also, the third order models can be used for studying phenomena that are not included in second order models. Third order models are usually much more computationally intensive than second order models, but they can still find periodic steady state solutions fast enough, that they are practical for numerical optimisation of complete machine designs. The state of the art in third order modelling of Stirling and related machines is currently represented by the only commercially available third order simulation program: *Sage*.

CFD is currently being used successfully for studying phenomena that cannot be studied using one-dimensional models. These phenomena include the effects of geometric details on multi-dimensional flow patterns, pressure losses, and temperature and heat transfer distributions. Current studies use either commercially available CFD packages, such as *Fluent* or *CFD-ACE+*, or in-house codes developed at universities and/or at institutions such as NASA.

CFD is very computationally intensive. Finding a solution to a complete engine model, where the temperature distribution in the regenerator matrix approaches periodic steady state, is presently very time consuming even on a parallel computer platform. This appears to make CFD impractical for optimising a complete machine layout using

numerical optimisation. It is my anticipation that it will take a number of years before this changes.

2.8. Classification of the present work

In this work the focus has been on simulation methods that are useful for design optimisation of complete machines. This presently appears to rule out multi-dimensional CFD as a viable option. Instead the work done here has been aimed at the high end of third order 1D modelling of Stirling machines and related machines.

(This page intentionally left blank)

3. Modelling of 1D, oscillating, compressible flow in reciprocating machines

This chapter presents the theory, implementation, and testing of an approach for modelling machines with oscillating, one-dimensional, compressible flow as initial value problems (IVPs) or as boundary value problems (BVPs).

The modelling approach was developed with the following design criteria:

- It should be able to model Stirling machines and similar machines with good accuracy.
- It should produce models to which solutions can be found quickly enough that the models are practical to use for design optimisation.
- It should produce models where the equations are directly accessible so that they are easy to modify.
- The equations should be as easy as possible to understand and modify by someone with a background in engineering thermodynamics.

The design criterias are deliberately written in soft terms in order to reflect the way the design criterias were used during this work: I kept the above design criterias in mind and then made choices of methods based on personal preferences.

In this chapter the modelling approach is presented as an approach for modelling the machines as IVPs. In chapter 4 it is then discussed how periodic BVPs can be solved.

The point is that the equations do not, for instance, use harmonic series representations of heat fluxes which would only make sense for periodic solutions; the equations are valid both for both transient and steady state problems. The equations are also not tied into a single numerical method.

The modelling approach was developed for making models of Stirling machines and pulse tube coolers. But the modelling approach is general in nature and is not limited to these specific types of machines.

This chapter begins with section 3.1 which describes the reasoning behind the basic choice of discretisation method for the governing equations for fluid flow. Section 3.2 describes the theory of the modelling approach and its methods for enhancing precision and for reducing the computational efforts needed to perform simulations.

Section 3.3 describes how the modelling approach has been implemented in this work as models in a simulation package called *MUSSIM*. The basic structure of the *MUSSIM* software, the structure of the models, and some performance enhancing features of the software are described.

In section 3.4 a Stirling machine model is briefly described. The model has been used for testing the performance enhancing features of the *MUSSIM* software and for testing the modelling approach in combination with an IVP solver. An overview of the tests is presented in section 3.5, and the sections 3.6 through 3.9 contains descriptions and results for the individual tests.

The contents of this chapter are summarised in section 3.10.

3.1. Choice of discretisation method for the governing equations

This work has been focussed on third order modelling of Stirling machines and related machines, i.e. focus has been on models where the governing equations for fluid flow are discretised in one space dimension and time. The equations of such models represent a form of partial differential equations.

Commiso (1994) discussed different methods for discretising and solving partial differential equations in the context of Stirling machine modelling. The discussion included finite difference methods, finite volume methods, finite element methods, spectral methods, the method of characteristics, and the method of lines. Commiso concluded that none of the methods had proven to be clearly superior to the others.

Organ (1992), on the other hand, strongly advocated the method of characteristics. The method of characteristics has the advantage that the computed solutions are free from numerical diffusion so that discontinuities in pressure or temperature can be maintained if they travel in a flow field where no diffusion is present. Organ vehemently criticised the use of discretisation schemes based on Eulerian grids, i.e. schemes where gas moves relative to the nodes or volumes, for modelling Stirling machines. According to Organ such schemes produce solutions where the rates of diffusion and the rates of pressure information propagation are functions of the discretisation of space and time. Organ suggested that refining the discretisation for such finite difference or finite volume methods merely leads to different solutions but not necessarily to more accurate solutions.

Organs criticism of methods based on Eulerian grids was rebutted by Bauwens (1993a, 1993b). Bauwens argued that if one uses a convergent numerical scheme, Eulerian or not, then minimising the errors due to numerical diffusion to a tolerable level is merely a question of using a sufficiently fine grid. Organs broad criticism of the lack of convergence for methods based on Eulerian grids also appears to conflict with the mathematical derivation of the speed of convergence for different finite difference schemes given by authors such as Sod (1985), with the good results achieved with finite different methods for high speed gas dynamics by for instance Adewumi, Eltohami and Ahmed (2000), and with the performance of the widely adopted finite volume methods in CFD codes.

Kühl (1990) and Kühl and Schultz (1995) emphasize that using a discretisation on a Eulerian grid has practical advantages. When models are formulated using a discretisation on a Eulerian grid it is easy to separate the implementation of the equations from the implementation of the numerical method that is used for solving the equations; this separation is difficult to achieve if, for instance, the method of characteristics is used. Using a Eulerian discretisation further has the advantage, that it is relatively easy to keep the implementations of equations for different components separate. This makes it easier to make a modular implementation where blocks of equations for different types of components are stored separately.

If one chooses to use a discretisation on a Eulerian grid then there appears to be two main approaches for formulating one-dimensional models:

1. To discretise the differential forms of the one-dimensional governing equations for fluid flow, i.e. the equations for mass, energy, and momentum, using a finite difference scheme.
2. To apply the integral form of the governing equations for fluid flow to a string of control volumes.

The first approach has been used in, for instance, *GLIMPS* and *Sage* by Gedeon (1986, 1994) and *MS*2* by Bauwens (1993a). The second approach, which from here on is denoted as the control volume approach, was used in *HFAST*. The control volume approach is also used, for instance, in the simulation program by Carlsen (1993) (A detailed description of the simulation program by Carlsen has never been published, but the program has been used successfully for designing several Stirling engines).

If a conservative discretisation scheme for the differential form of the equations is chosen then the equations that result from these two formulations are very similar (Commiso, 1994). One difference between the two approaches is that the integral form of the governing equations for fluid flow is meaningful regardless of the shapes of the control volumes and how the volumes are connected. When discretisation of the differential forms of the equations is performed, then mathematical attention must be paid to, for instance, flow area discontinuities in order to maintain a meaningful formulation (Bauwens, 1993b).

There appears to be no fundamental reasons to universally prefer one of the two forms of the governing equations for fluid flow over the other.

In this work it has been chosen to make a control volume based formulation based on a Eulerian grid. The main reasons behind this choice are:

1. A Eulerian grid of control volumes is simple to understand and work with for a person with a background in engineering thermodynamics. Understanding a model is very important when creating or modifying the model.
2. It is easy to completely separate models from the numerical methods that are used for solving them. This has the advantage that one can verify the correctness of the applied numerical methods by comparing the results from different numerical methods applied to the same problem. One is also free to change the numerical method if one becomes aware of a better method than the one presently used.
3. Equations for different components can easily be separated in the implementation. This makes it relatively easy to add extra components to a model by adding extra blocks of equations for the components.

By choosing a control volume based formulation on a Eulerian grid it has been accepted that numerical diffusion will be present in the solutions. The formulation will also have difficulties in tracing discontinuities which propagate relative to the grid of control volumes. But methods exist for minimising numerical diffusion and such methods are discussed below.

3.2. The modelling approach in theory

This section presents the basic theory of the modelling approach. Features for enhancing the accuracy of solutions and for reducing the computational effort needed to simulate Stirling machines are also discussed.

Parts of the contents of this section have previously been published (Andersen, Carlsen and Thomsen, 2004). Compared to this paper it is primarily the section "*Interpolation of state variables and velocities to control volume boundaries*" that has been revised and expanded. The model formulation has also been updated so that kinetic energy is treated with better consistency. The section "*Discretisation of the solids*" was not included in the paper.

3.2.1. Discretisation of the gas domain on a staggered mesh of control volumes

The modelling approach presented here is based on the integral form of the conservation equations for mass, energy, and momentum applied to a staggered mesh of two overlapping strings of control volumes. Conservation of mass and energy is applied to one string and conservation of momentum is applied to the second string. The control volumes, to which conservation of momentum is applied, are located so that their centres coincide with the boundaries of the control volumes where conservation of mass and energy are applied. This is the common approach of the finite volume method of CFD. The staggered mesh approach ensures direct coupling between pressures, which are determined by the mass and energy balances, and velocities, which are determined by the momentum balances. This direct coupling yields stable and bounded solutions.

Fig. 6 illustrates the staggered mesh of control volumes. The solid line control volumes are used for the mass and energy balances and the dashed line control volumes are used for the momentum balances.

From the balance equations for these control volumes a system of first order ordinary differential equations (ODEs) is derived. ODEs for variables that, as explained later, ultimately represent the pressures and the temperatures in the control volumes are derived from the mass and energy balances for the solid line control volumes. ODEs for the velocities in the domain are derived from the momentum balances for the dashed line control volumes.

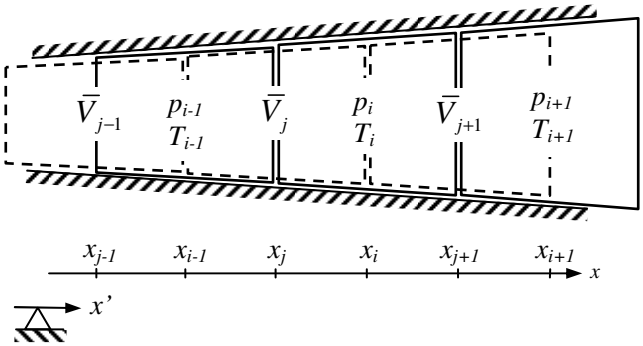


Fig. 6. A staggered mesh of control volumes

3.2.2. The mass and energy balances

The mass balance for the solid line control volume with index *i* is written as follows:

$$\frac{dm_i}{dt} = \dot{m}_j - \dot{m}_{j+1} = \underbrace{\rho_j \cdot \bar{V}_j \cdot A_{c,j} - \rho_{j+1} \cdot \bar{V}_{j+1} \cdot A_{c,j+1}}_{Advection} \tag{1}$$

When writing the energy balance for control volume *i* it is assumed that gravitational potential energies are of negligible magnitudes so that the specific total energy at the centre of the control volume is $e_i = u_i + \frac{\bar{V}_i^2}{2}$:

$$\frac{dE_i}{dt} = \frac{d(m_i \cdot e_i)}{dt} = \underbrace{\dot{m}_j \cdot \left(h_{gas,j} + \frac{\bar{V}_j^2}{2} \right) - \dot{m}_{j+1} \cdot \left(h_{gas,j+1} + \frac{\bar{V}_{j+1}^2}{2} \right)}_{Advection} + \underbrace{-k_{eff,j} \cdot A_{c,j} \cdot \left. \frac{dT}{dx} \right|_j + k_{eff,j+1} \cdot A_{c,j+1} \cdot \left. \frac{dT}{dx} \right|_{j+1}}_{Axial\ conduction\ including\ turbulent\ conduction\ enhancement} \underbrace{\sum_k h_{conv,i,k} \cdot A_{ht,i,k} \cdot \left(T_{w,i,k} - T_i \right)}_{Convective\ heat\ transfer\ from\ walls} - \underbrace{p_i \cdot \frac{dV_i}{dt}}_{Volume\ change} \quad (2)$$

The values of h_{conv} and k_{eff} must be calculated using empirical correlations.

Additional terms that affect the energies in the control volumes, such as internal heat production, can be included in (2) as additional terms if needed. The gas in volume *i* may exchange heat with more than one surface.

3.2.3. Transformation of equations for mass and energy balances

Models can be formulated directly using the ODEs for the mass and energy balances in (1) and (2) but doing so may not be optimal. If the computational domain contains control volumes of different sizes then the masses and energies in the volumes will also differ in sizes, even if the pressure and temperature are constant throughout the domain. If the sizes of the control volumes differ by orders of magnitudes then the corresponding masses and energies will also differ by orders of magnitude. This can make it more difficult to accurately solve the ODEs for the masses and energies due to bad scaling of the variables.

A transformation of the ODEs for masses and energies into ODEs for pressures and temperatures can be desirable because it can improve the scaling of the variables and at the same time make it easier for an engineer to provide realistic initial values to a model. The transformation can also help to separate fast and slow phenomena into different variables because changes in pressure will propagate at sonic speeds while major changes in temperature generally travel with the, usually much slower, bulk flow speed induced by the motion of one or more pistons. This separation can make it easier and faster to accurately solve the ODEs.

For an ideal gas, using $p \cdot V = m \cdot R \cdot T$, $U = m \cdot u$, and $du = c_v(T) \cdot dT$, gives:

$$\begin{aligned} \frac{dT_i}{dt} &= \frac{1}{m_i \cdot c_v(T_i)} \cdot \left(\frac{dU_i}{dt} - \frac{U_i}{m_i} \cdot \frac{dm_i}{dt} \right) \\ \frac{dp_i}{dt} &= p_i \cdot \left(\frac{1}{m_i} \cdot \frac{dm_i}{dt} + \frac{1}{T_i} \cdot \frac{dT_i}{dt} - \frac{1}{V_i} \cdot \frac{dV_i}{dt} \right) \end{aligned} \quad (3)$$

The transformation defined by (3) needs $\frac{dm_i}{dt}$ and $\frac{dU_i}{dt}$ as input, and these derivatives must be extracted from (1) and (2). $\frac{dm_i}{dt}$ is given explicitly by (1). For Stirling machines it will usually be safe to neglect kinetic energy (Bauwens, 1993a), so that one can obtain $\frac{dU_i}{dt}$ by simply assuming that $\frac{dE_i}{dt} = \frac{dU_i}{dt}$.

The transformation can be enhanced by scaling the units of the pressures and temperatures between the model and the numerical method. Eq. (3) requires a consistent set of units, but the numerical method may integrate the pressures and temperatures in units of, say, MPa and hK (hekto-Kelvin) if this helps to achieve better scaling of the variables.

When it is not desired to assume ideal gas or to neglect kinetic energy then an alternative approach, which is much more general with respect to the equation of state for the gas, can be used to combine thermodynamic properties into variables that behave like pressure and temperature. For many applications the pressures, temperatures, and velocities will be so that the specific internal energy u, and also the specific total energy e, depends mostly on temperature. The energy density $\rho \cdot e$ on the other hand will almost cancel out the individual dependencies of ρ and e on temperature if u is close to directly proportional to the temperature. A suitable linear translation of u can be necessary to eliminate the influence of the reference temperature for u in the gas property functions if this reference temperature differs from 0 K. When $\rho \cdot e$ is nearly independent of temperature it will behave much like a pressure; it even has units of pressure.

To obtain good scaling of variables ρ and e can either be made dimensionless using suitable linear translations or just be scaled by using appropriate units. The transformations from mass and energy into density and energy density can be written with (1) and (2) as input as:

$$\begin{aligned} \frac{de_i}{dt} &= \frac{1}{m_i} \cdot \left(\frac{dE_i}{dt} - \frac{E_i}{m_i} \cdot \frac{dm_i}{dt} \right) \\ \frac{d\rho_i}{dt} &= \frac{1}{V_i^2} \cdot \left(V_i \cdot \frac{dm_i}{dt} - m_i \cdot \frac{dV_i}{dt} \right) \\ \frac{d(\rho_i \cdot e_i)}{dt} &= \frac{d\rho_i}{dt} \cdot \frac{E_i}{m_i} + \frac{m_i}{V_i} \cdot \frac{de_i}{dt} \end{aligned} \tag{4}$$

Specific energy and energy density may not be as easy to relate to as temperature and pressure when manually making initial guesses for a solution. But at least these variables are independent of the sizes of the control volumes and hence they make it easier to make initial guesses that correspond to uniform conditions in regions of a machine. It is, of course, also possible to automate the calculation of initial guesses in a model. When this is done the disadvantages of the transformation (4) compared to the transformation (3) are reduced, while the advantages of greater generality with respect to the equations of state for the gas and the treatment of kinetic energy are preserved.

3.2.4. The momentum balance

When modelling Stirling machines it is not uncommon to simplify the momentum balance by neglecting the inertia of the gas. This approach was also pursued in the beginning of this work as documented in the accompanying *Paper A*. However, it appeared desirable to include the inertia terms in the momentum equation in order to eliminate the risk that their exclusion caused important effects to be neglected.

Experiments were carried out which compared the computational efforts needed to compute solutions with and without the inertia terms in the momentum equation. These experiments indicated that the inertia terms caused only little overhead for

periodic steady state solutions. It was also found that the iterations performed in a semi-implicit Runge-Kutta method during simulations actually converged better during flow reversals when the inertia terms were included than when they were neglected. There was, however, a measurable overhead due to the inertia terms when simulations were performed from non-smooth initial pressure distributions. But it was found that the overhead could be reduced to a tolerable level by using an appropriate method for dissipating acoustic waves.

On this background it was chosen to include the full momentum balance in the modelling approach.

The momentum balance for the dashed line control volume *j* is written as (5).

$$\begin{aligned}\frac{d\left(m_j \cdot \bar{V}_j\right)}{dt} = \; & \underbrace{p_{i-1} \cdot A_{c,i-1} - p_i \cdot A_{c,i} + \tilde{p}_j \cdot \left(A_{c,i} - A_{c,i-1}\right)}_{Pressure\ forces} + \\ & \underbrace{\left(\rho_{i-1} \cdot \bar{V}_{i-1}{}^2 \cdot A_{c,i-1} - \rho_i \cdot \bar{V}_i^2 \cdot A_{c,i}\right)}_{Advection} - \underbrace{\frac{\bar{V}_j}{\left|\bar{V}_j\right|} \cdot \left|F_{wall,j}\right|}_{Wall\ friction} + \\ & \underbrace{F_{AD,j}}_{Artificial\ dissipation} - \underbrace{m_j \cdot \frac{d\bar{V}_{CS}}{dt}}_{Acceleration\ of\ coordinate\ system}\end{aligned} \tag{5}$$

By expanding the accumulation term on the left hand side in (5) the time derivative of the velocity in volume *j* becomes:

$$\begin{aligned}\frac{d\bar{V}_j}{dt} = \; & \frac{1}{m_j} \cdot \Bigg\{ \underbrace{p_{i-1} \cdot A_{c,i-1} - p_i \cdot A_{c,i} + \tilde{p}_j \cdot \left(A_{c,i} - A_{c,i-1}\right)}_{Pressure\ forces} + \\ & \underbrace{\left(\rho_{i-1} \cdot \bar{V}_{i-1} \cdot A_{c,i-1} \cdot \left(\bar{V}_{i-1} - \bar{V}_j\right) - \rho_i \cdot \bar{V}_i \cdot A_{c,i} \cdot \left(\bar{V}_i - \bar{V}_j\right)\right)}_{Advection\ and\ accumulation} \\ & \underbrace{- \frac{\bar{V}_j}{\left|\bar{V}_j\right|} \cdot \left|F_{wall,j}\right|}_{Wall\ friction} + \underbrace{F_{AD,j}}_{Artificial\ dissipation} \Bigg\} - \underbrace{\frac{d\bar{V}_{CS}}{dt}}_{Acceleration\ of\ coordinate\ system}\end{aligned} \tag{6}$$

In the following the term $\Delta p_{f,j}$ is used to represent the absolute value of the pressure difference due to flow friction between the inlet and outlet of volume *j*. $\Delta p_{f,j}$ can be calculated using empirical correlations for the friction factor and loss coefficients. Correlations developed for steady state flow are often readily available and can be used as an approximation. The term $F_{wall,j}$ in (6) can then be approximated by multiplying the pressure loss with a cross sectional reference area $A_{c,ref,j}$:

$$F_{wall,j} = \Delta p_{f,j} \cdot A_{c,ref,j} \tag{7}$$

It appears appropriate to choose $A_{c,ref,j}$ to be equal to either $A_{c,i-1}$, $A_{c,i}$, or some mean value in between these two cross sectional areas. To determine the pressure $\tilde{p}_j$ in (6) acting on the area difference between the inlet and outlet of volume *j*, we impose the condition that (6) must match the energy equation for steady state incompressible flow

in a stationary tube, i.e. the extended Bernoulli equation, between x_{i-1} and x_i when gravitational potential energies are neglected:

$$\left(p_{i-1} + \rho_j \cdot \frac{\bar{V}_{i-1}{}^2}{2}\right) - \left(p_i + \rho_j \cdot \frac{\bar{V}_i^2}{2}\right) = \frac{\bar{V}_j}{\left|\bar{V}_j\right|} \cdot \Delta p_{f,j} \tag{8}$$

It can be verified by insertion that this is achieved if the pressure $\tilde{p}_j$ is calculated using either (9) or (10) and the artificial dissipation force is zero. Note that the last terms in (9) and (10) simplify greatly if $A_{c,ref,j}$ is chosen equal to either $A_{c,i-1}$ or $A_{c,i}$.

$$\tilde{p}_j = p_i + \frac{1}{2} \cdot \rho_j \cdot V_i^2 \cdot \left(1 - \frac{A_{c,i}}{A_{c,i-1}}\right) + \frac{\frac{A_{c,ref}}{A_{c,i-1}} - 1}{\frac{A_{c,i}}{A_{c,i-1}} - 1} \cdot \frac{\bar{V}_j}{\left|\bar{V}_j\right|} \cdot \Delta p_{f,j} \tag{9}$$

$$\tilde{p}_j = p_{i-1} + \frac{1}{2} \cdot \rho_j \cdot \bar{V}_{i-1}{}^2 \cdot \left(1 - \frac{A_{c,i-1}}{A_{c,i}}\right) - \frac{1 - \frac{A_{c,ref,j}}{A_{c,i}}}{1 - \frac{A_{c,i-1}}{A_{c,i}}} \cdot \frac{\bar{V}_j}{\left|\bar{V}_j\right|} \cdot \Delta p_{f,j} \tag{10}$$

Eqs. (9) and (10) give $\tilde{p}_j$ as the pressure in a solid line control volume plus two correction terms. At flow area discontinuities, such as the location where a tube connects to a large chamber, the correction terms containing the square of the velocity can cause instability. To avoid instability the choice between (9) and (10) should be made so that the squared velocity is as small as possible, i.e. so that the extrapolation from the pressure in the solid line control volume is as small as possible. In practice this can be done by using (9) where $A_{c,i} \geq A_{c,i-1}$ and using (10) elsewhere.

3.2.5. Artificial dissipation

For some machines it is known that the periodic steady state performance will be practically unaffected by acoustic waves. Acoustic waves are pressure waves that propagate with the speed of sound. In Stirling machines the acoustic waves typically oscillate at frequencies significantly higher than the operating frequency of the machines.

When a simulation is started from a bumpy pressure distribution then acoustic waves may be induced. As the simulation progresses the acoustic waves will eventually ring out. But due to the tiny time steps needed to track the acoustic waves during the initial phase of such a simulation the acoustic waves can cause a very significant and unwanted computational overhead when modelling Stirling machines.

An artificial dissipation force $F_{AD,j}$ is included in the modelling approach and this dissipation force can be used in such cases to accelerate the dissipation of the acoustic waves. I use (11) for calculating the artificial dissipation force.

$$F_{AD,j} = \left\{ -\tilde{\mu}_1 \cdot l_j \cdot \left(A_{c,i-1} \cdot \left. \frac{\partial \bar{V}}{\partial x} \right|_{i-1} - A_{c,i} \cdot \left. \frac{\partial \bar{V}}{\partial x} \right|_{i} \right) + \tilde{\mu}_2 \cdot l_j^{\,3} \cdot \left(A_{c,i-1} \cdot \left. \frac{\partial^3 \bar{V}}{\partial x^3} \right|_{i-1} - A_{c,i} \cdot \left. \frac{\partial^3 \bar{V}}{\partial x^3} \right|_{i} \right) \right\} \cdot p_{avrg} \tag{11}$$

The terms in (11) proportional to the first derivative of the velocity can be considered similar to a viscous normal stress (Bird, Stewart, and Lightfoot, 1960). The terms proportional to the third derivatives of the velocity cannot easily be related to physical effects.

The terms proportional to the third spatial derivatives of the velocity inhibit oscillations between neighbouring pairs of control volumes, and hence dampen oscillations with a short wavelength compared to the length of the control volumes. The terms proportional to the first spatial derivatives of the velocity penalize curved velocity profiles and hence dampen oscillations with longer wavelengths.

The artificial dissipation force is scaled by an average pressure that should be representative for the region where the artificial dissipation is applied. In this work I have used the instantaneous average pressure in the cylinder volumes of the machines for the scaling.

Care must be taken not to apply the artificial dissipation forces so that they interfere significantly with the mean solution. The magnitudes of the artificial dissipation coefficients $\tilde{\mu}_1$ and $\tilde{\mu}_2$ must be large enough to yield the desired smoothing effect and yet small enough to not significantly affect the mean solution. At the same time (11) should not be used where flow area discontinuities, localised pressure losses, or other localised phenomena affect the calculation of the velocity derivatives in (11). I have not observed problems from simply switching the artificial dissipation terms off, when any of the dashed line control volumes containing the velocities used for calculating the derivatives in (11) contain flow area changes or are prescribed localised pressure losses to model inlets, tube bends, or similar.

Visual inspection of solutions was used to check if the desired smoothing effect was achieved. Inspection of the ratio $\left| F_{AD,j} / F_{wall,j} \right|$ and experimentation with the artificial dissipation coefficients was used to test that the artificial dissipation did not interfere significantly with the mean solution. For simulations of Stirling engines, Stirling coolers, and pulse tube coolers with helium, nitrogen, air, and carbon dioxide at pressures in the range of 1-10 MPa I found the values $\tilde{\mu}_1 = 1.0 \cdot 10^{-4} s/m$ and $\tilde{\mu}_2 = 2.0 \cdot 10^{-6} s/m$ to be suitable.

3.2.6. Interpolation of state variables and velocities to control volume boundaries

The pressures p_j and enthalpies $h_{gas,j}$ (or alternately the temperatures T_j) at the boundaries of the solid line control volumes, and the velocities $\bar{V}_i$ at the boundaries of the dashed line control volumes are not given as explicit output by the modelling formulation. But these unknown values are needed, both explicitly and in gas property calculations, in the equations (1) through (10). Since the model formulation does not yield these values directly they must be approximated from known values in the control volumes next to the boundaries using some kind of interpolation method. The choice of

interpolation method turns out to be very important for the accuracy of the modelling approach.

Problems related to interpolation methods
The enthalpies at the boundaries of the solid line control volumes are needed for computing the amount of energy transported by advection between neighbouring solid line control volumes.

Unfortunately the obvious approach of using a symmetric interpolation method to compute the temperatures or enthalpies at boundaries does not work well because it will introduce strong non-physical temperature oscillations into solutions. The source of the problem with symmetric interpolation of boundary temperatures can be thought of as temperature information being able to travel against the flow direction. With symmetric interpolation the temperature at a boundary will depend as much on the temperature in the receiving control volume as on the temperature in the control volume where the mass flow is coming from.

The problem with non-physical oscillations can be avoided by using the so called upstream approximation. When using the upstream approximation it is assumed, that gas flowing across a boundary between two control volumes has a temperature equal to the average temperature in the control volume where the gas is coming from.

Unfortunately the upstream approximation introduces an artificial energy transport mechanism, known as numerical diffusion, into solutions when mass is advected along a temperature gradient. When mass flow is in the direction of decreasing temperature the temperatures at the boundaries between control volumes, and hence the enthalpy flows across the boundaries, are overestimated by the upstream approximation. When mass is flowing in the direction of increasing temperature the enthalpy flows at the boundaries are underestimated. The numerical diffusion has nothing to do with the underlying governing equations but is only an artefact of the spatial discretisation scheme.

With the upstream approximation the magnitude of the artificial energy transport due to numerical diffusion is proportional to the mass flow, the temperature gradient, and the length of the control volumes. So when the number of control volumes is doubled the magnitude of the error due to numerical diffusion is halved. The numerical diffusion can hence be said to be a first order error term. Numerical diffusion in models of Stirling machines is a well known problem (Gedeon, 1984)(Kühl and Schultz, 1995)(Commiso, 1994) that can be very severe because of the large axial temperature gradients in regenerators.

The problems with non-physical oscillations and numerical diffusion cause a need for a specialised interpolation method for the boundary enthalpies. The interpolation method must force upstream temperatures to take some precedence over downstream temperatures without introducing too much numerical diffusion when steep temperature gradients are present in the solution.

Background for choosing interpolation methods
When deciding which form of interpolations to use for approximating the unknown values one should balance three design criteria:

- The interpolations should increase the accuracy of models for predicting the performance of machines.

- The interpolations should not increase the computational cost of solutions too much. The computational cost of the interpolations themselves will usually not be very significant. But if the interpolations depend on the local solution in a strongly non-linear way then the computational cost of integrating a model using an implicit integration method can increase significantly.
- The interpolations should not introduce too strong non-physical oscillations into solutions (these oscillations are discussed below).

When weighing these requirements one should also take into account the characteristics of the solutions to the models. For Stirling and similar machines the following characteristics are known from experience:

- Very strong temperature gradients exist in regenerators. The axial temperature profile in the regenerator is not shifted significantly back and forth in the flow direction, i.e. the temperature inside a regenerator is mainly a function of the axial space coordinate. Limiting numerical diffusion in regenerators is known to be critical to Stirling machine modelling.
- Propagating discontinuities in pressures and velocities do not occur in periodic steady state solutions. Discontinuities in pressures and velocities only occur at localised pressure losses and flow area discontinuities which are all fixed in space relative to the control volumes. In other words: Shocks do not form and propagate in periodic steady state solutions. However, shocks may form as a result of poor guesses for the initial values. Poor guesses for the initial values can, for instance, result from a divergence in a numerical method trying to find the initial values corresponding to a periodic steady state solution. Hence it is important that shocks can be dissipated and that they do not cause catastrophic instabilities that make it impossible to find a solution.
- Temperature discontinuities can, conceptually, be produced by flow reversals. Consider, for instance, a tube connected to a large cylinder volume and assume that gas is flowing out of the tube and into the cylinder volume with a temperature different from the temperature in the cylinder volume. In this case a temperature discontinuity can form at flow reversal when gas from the cylinder volume begins to flow into the tube. In a real machine, however, the turbulent mixing at the inlet/outlet should smooth away any would-be discontinuity. In a one-dimensional model one can achieve the same effect artificially by connecting components appropriately. But waves with steep temperature gradients may form at flow reversals in spite of the smoothing. Temperature discontinuities can still also occur as a result of poor initial values. Hence temperature discontinuities or steep wave fronts must not make the equation system unsolvable.

Appropriate interpolation methods

Both interpolation methods that depend linearly on the local solution and interpolation methods that depend on the local solution in non-linear ways have been considered in this work. Both classes of methods can minimise numerical diffusion to tolerable levels. The methods that depend linearly on the local solution generally yield the fastest simulations but produce solutions which can contain non-physical oscillations. The methods that depend on the local solution in a non-linear way minimise both numerical diffusion and non-physical oscillations but they tend to make simulations more computationally expensive.

Linear interpolation methods

Kühl (1990) presented an interpolation method based on a blend between symmetric linear interpolation and linear extrapolation through two upstream points. The optimum weight factors were determined by Kühl to be 0.3 for the extrapolation and 0.7 for the interpolation. Except for its dependence on the flow direction the approach by Kühl depends only linearly on the local solution.

I have previously recommended an asymmetric interpolation method based on a cubic polynomial that goes through one upstream point and two downstream points and whose first derivative at the upstream point equals the slope of the tangent through two upstream points. Except for its dependence on the flow direction this method also depends linearly on the local solution. The method was previously described by Andersen, Carlsen, and Thomsen (2004).

In practice, however, the differences in simulation results between the asymmetric cubic interpolation and the asymmetric interpolation method by Kühl are only slight. Also the interpolation method based on the asymmetric cubic polynomials has the disadvantages that is it more difficult to implement when using a non-uniform mesh and that the factor by which upstream information is given precedence cannot be adjusted. For these reasons this author now prefers the method by Kühl over the asymmetric cubic interpolation method.

The method by Kühl can be modified slightly by using cubic polynomials for the interpolation instead of linear interpolation. When the interpolation method is used at locations where oscillations are well damped by physical mechanisms, so that a small weight factor for the upstream extrapolation can be used, then using cubic interpolation instead of linear interpolation might result in better accuracy. In the following this modified method will be denoted as *CILE* for *Cubic Interpolation Linear Extrapolation*. The CILE method also depends only linearly on the local solution.

Fig. 7 illustrates some of the advantages and problems with methods such as the CILE method and the upstream approximation. The figure shows snapshots at two different times of a temperature front travelling from left to right through a gas filled tube, which was discretised into 96 control volumes of uniform size. The front was created with a step input of cold gas being blown into the left end of the tube. The upstream approximation and the CILE method with three different weight factors for the extrapolation term were used for interpolating enthalpies at the boundaries of the control volumes.

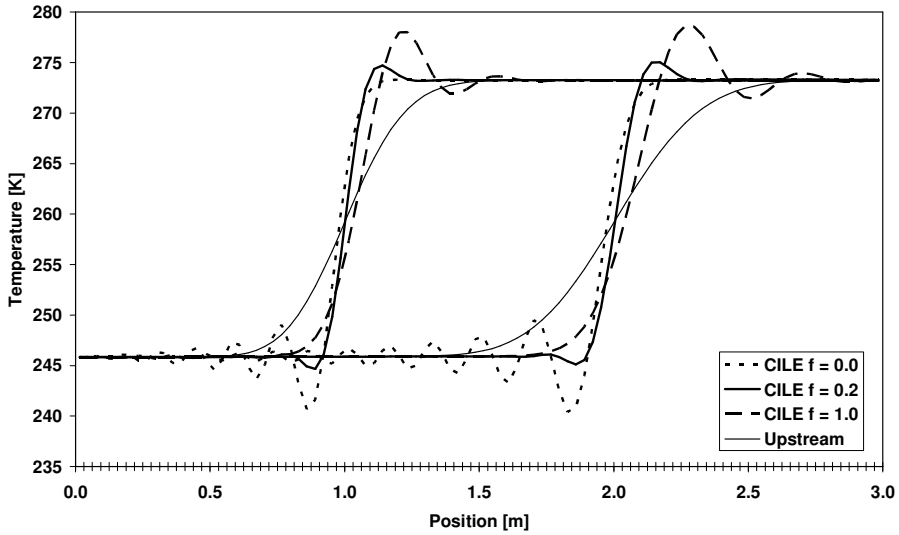


Fig. 7. A travelling temperature front in a 3 m long tube divided into 96 solid line control volumes. Boundary enthalpies were computed using the upstream approximation and the CILE method, i.e. a blend between linear upstream extrapolation (weight factor f) and cubic interpolation (weight factor 1-f). The tick marks on the horizontal axis indicate the positions of the boundaries between the solid line control volumes.

Fig. 7 shows that the upstream approximation caused the step input to be smeared by numerical diffusion. The gradient of the smeared temperature front also diminished visibly between the two snapshots.

The three curves for the CILE method with different weight factors (0, 0.2, and 1) for the linear extrapolation in Fig. 7 show, that the CILE method also caused smearing of the step input and that it also caused non-physical oscillations. But the temperature gradient at the front was steeper than for the upstream approximation and the gradient was preserved quite well between the two snapshots. This means that the numerical diffusion was smaller than for the upstream approximation.

Fig. 7 shows that using only symmetric interpolation (CILE f = 0.0) caused strong oscillations to trail the front. Using only extrapolation (CILE f = 1.0), on the other hand, caused an overshoot and strong oscillations to precede the front. In the latter case the oscillations propagated faster than the bulk flow velocity which transported the temperature front. Using a weight factor of 0.2 for the linear extrapolation appeared to be a good compromise for the CILE method when balancing the overshoot and the accompanying preceding oscillations with the trailing oscillations. But even with the optimised weight factor the non-physical oscillations were still visible.

Non-linear interpolation methods

The fluid dynamics literature contain a number of different methods, such as flux limiters (Suratanakavikul and Marquis, 1999) or flux splitting (Fedkiw et al, 1998), that depend on the instantaneous solution in different non-linear ways. These methods can both minimise numerical diffusion and non-physical oscillations but at the cost of increasing the computational efforts needed to perform simulations.

By experimentation is has been found that some flux limiters only cause a slight overhead relative to the CILE method. A flux limiter can be perceived as a function Ψ_j that determines the weight factor for the values in the neighbouring control volumes when interpolating a value at a boundary. The flux limiter function depends on a sensor variable which is related to the smoothness of the local solution. In this work flux

limiters have been implemented with a sensor variable, r_j, based on consecutive gradients. If the enthalpy is to be interpolated then r_j is calculated as follows, where ε is a very small number:

$$r_j = \begin{cases} sign[(h_{i-1} - h_{i-2}) \cdot (h_i - h_{i-1})] \cdot \dfrac{\max(\varepsilon, |h_{i-1} - h_{i-2}|)}{x_{i-1} - x_{i-2}} \Big/ \dfrac{\max(\varepsilon, |h_i - h_{i-1}|)}{x_i - x_{i-1}}, \overline{V}_j \geq 0 \\ sign[(h_i - h_{i+1}) \cdot (h_{i-1} - h_i)] \cdot \dfrac{\max(\varepsilon, |h_i - h_{i+1}|)}{x_i - x_{i+1}} \Big/ \dfrac{\max(\varepsilon, |h_{i-1} - h_i|)}{x_{i-1} - x_i}, \overline{V}_j < 0 \end{cases} \quad (12)$$

The sensor variable will hence be equal to 1 if the gradient of the solution is constant. One flux limiter found to cause only moderate overhead is due to van Leer (Suratanakavikul and Marquis, 1999) and is defined in the following way:

$$\Psi_j = \frac{r + |r|}{1 + r} \quad (13)$$

The interpolation of the enthalpy at the boundary using the flux limiter function is then carried out as follows:

$$h_j = \begin{cases} h_{i-1} + \dfrac{\Psi_j}{2} \cdot (h_i - h_{i-1}), \overline{V}_j \geq 0 \\ h_i + \dfrac{\Psi_j}{2} \cdot (h_{i-1} - h_i), \overline{V}_j < 0 \end{cases} \quad (14)$$

Fig. 8 shows the same temperature front as Fig. 7 from simulations using:

- The CILE method with a weight factor of 0.2 for the linear extrapolation.
- The asymmetric cubic interpolation method.
- The method by Kühl.
- The van Leer flux limiter.

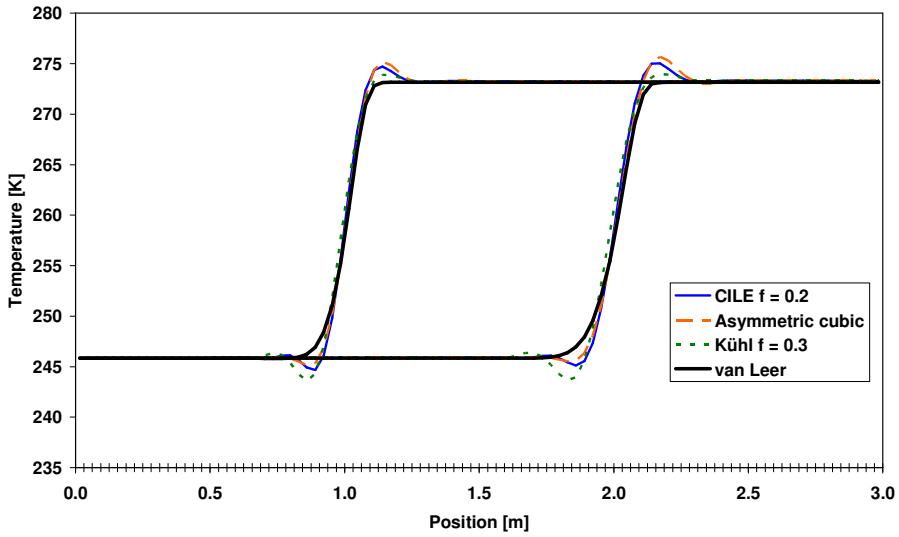


Fig. 8. A travelling temperature front in a 3 m long tube divided into 96 solid line control volumes. Boundary enthalpies were computed using: 1. The CILE method with a weight factor of 0.2 for the upstream extrapolation, 2. asymmetric cubic interpolation, 3. the interpolation method by Kühl, 4. the flux limiter by van Leer. The tick marks on the horizontal axis indicate the positions of the boundaries between the 96 solid line control volumes in the tube.

In Fig. 8 the results for the three interpolation methods which depend linearly on the local solution are very similar. These methods cannot maintain the temperature discontinuity but they maintain a slightly smeared temperature gradient well between the two snapshots. All three methods have overshoots and oscillations that appear to grow slowly in amplitude as the temperature front travels. For solutions where numerical diffusion must be minimised and where the oscillations will not grow to destroy the solution either of three interpolation methods appear to be more attractive than the upstream approximation.

Fig. 8 shows that the van Leer flux limiter both preserves the steep gradient and avoids the non-physical oscillations. For situations where avoiding non-physical oscillations is more important than achieving optimum simulation speed the van Leer flux limiter appears to be more attractive than the three interpolation methods which depend linearly on the local solution.

It appears to make little sense to use the specialised interpolation methods right next to flow area discontinuities where different components are connected. In that case, one of the points used for the interpolation would be located in the neighbouring component where conditions might be different. Hence the specialised interpolation methods are not applied when some of the points used for the interpolation would fall across flow area discontinuities.

Upstream extrapolation is applied to find enthalpies at flow area discontinuities, such as connections between components. If the upstream control volume is assumed to be perfectly mixed then the upstream extrapolation reduces to the standard upstream approximation.

Interpolation of pressures and velocities at boundaries

Intuitively, it appears to make sense to use the same specialised interpolation method for interpolating both pressures, temperatures and velocities; doing so applies the same upwind bias to advection of energy, mass, and momentum.

In periodic solutions to Stirling machine models, however, I have not observed that the choice of interpolation method for the pressure makes a significant difference. But pressure discontinuities travelling through a machine in a transient solution do tend to cause fewer oscillations when the CILE or van Leer methods are used for interpolating pressures. In this context it has been observed for the CILE method that using a larger weight factor, such as 0.5 instead of 0.2, for the extrapolation term seems most effective at reducing oscillations.

In this work no advantages in using asymmetric interpolation methods for the velocities have been observed for Stirling machine models. Hence symmetric cubic polynomials have been used for interpolating velocities to the boundaries of the dashed line control volumes.

3.2.7. Discretisation of the solids

The temperatures of the solids in a machine must also be modelled. To do so the solids are discretised into control masses. The discretisation is made so that the axial distribution of the control masses matches the axial distribution of the solid line control volumes, i.e. so that the boundaries between the control masses match the boundaries between the solid line control volumes.

The solids corresponding to each solid line control volume may be divided into multiple control masses. In an annular regenerator, for instance, there would typically be three control masses for each solid line control volume; two control masses would the represent the inner and outer walls of the annular duct and the third control mass would represent the regenerator matrix.

Depending on the assumptions made about the temperature distributions within the control masses there are different methods for modelling the temperatures of the control masses.

One approach is to assume that the temperatures are uniform within the control masses, i.e. to use a lumped formulation for the temperatures of the control masses. This assumption is often accurate for the control masses representing a regenerator matrix, which is made from very fine wires. Wire diameters in regenerators are typically between 20-100 μm and in this work the corresponding Biot numbers have at most been of the order of magnitude 10^{-2}. In this case the ODE for the temperature of the control mass can be derived from an energy balance:

$$\begin{aligned} \frac{dU_{w,i,k}}{dt} &= m_{w,i,k} \cdot c_{v,w,i,k}(T) \cdot \frac{dT_{w,i,k}}{dt} \\ &= -h_{conv,i,k} \cdot A_{ht,i,k} \cdot (T_{w,i,k} - T_i) - \sum conduction\ in\ solids \\ &\Downarrow \\ \frac{dT_{w,i,k}}{dt} &= -\frac{h_{conv,i,k} \cdot A_{ht,i,k} \cdot (T_{w,i,k} - T_i) + \sum conduction\ in\ solids}{m_{w,i,k} \cdot c_{v,w,i,k}(T)} \end{aligned} \tag{15}$$

Another approach is to assume that the wall temperatures are constant. This assumption can be accurate for walls where the surface heat fluxes are relatively small. If the oscillations in the surface temperatures are small compared to the temperature differences between the gas and the wall, then the surface temperature oscillations may have only a minor relative influence on the calculated heat transfer between the gas and the wall.

The constant temperature of a control mass at periodic steady state can be determined from the integral condition (16) which specifies that the net heat input to the control mass during one cycle must be zero. The integrated heat input must include both the heat exchange between the gas and the control mass, and the heat input to the control mass from conduction in the solids.

$$0 = \int_{t_0}^{t_0+\Delta t} \left[h_{conv,i,k} \cdot A_{ht,i,k} \cdot (T_{w,i,k} - T_i) + \sum Conduction\ in\ solids \right] dt \tag{16}$$

It is also possible to use a general discretisation of the solids in the radial direction in order to model the heat conduction inside the walls and thereby be able to predict the oscillations in the surface temperatures. This approach, however, is unattractive with respect to keeping simulations fast, because the approach will increase the number of equations.

An alternative approach is to begin from an analytical solution for the temperature oscillations inside a solid slab when the slab is subjected to a harmonically oscillating heat flux. One can then design specialised coarse discretisations of a wall so that the solution from the coarse discretisation will approach the analytical solution for

the solid slab when the surface heat flux is harmonic. Kühl (1990) presented such an approach for use in cylinder volumes, where the first harmonic of the wall temperature oscillations could be accounted for by using only one extra equation pr. wall segment.

In this work it has been chosen to use the lumped approach (15) to model regenerator matrix temperatures and to assume that all other wall temperatures are constant.

3.2.8. Conditions for periodic steady state

Models built using the modelling approach described above will contain the following variables that exhibit periodic behaviour according to the definition in section 1.1:

- Specific energy densities $\underline{\rho \cdot e}$ (or pressures $\underline{p}$) in the gas.
- Specific energies $\underline{e}$ (or temperatures $\underline{T}$) in the gas.
- Gas velocities $\underline{\bar{V}}$.
- Regenerator matrix temperatures $\underline{T_m}$.

Models will also contain constant wall temperatures which can be thought of as parameters for each simulation.

In the following we collect the periodic variables in the vector $\underline{y_p}$. We denote the integration results from the right hand sides of the integral conditions for the wall temperatures (16) as the vector $\underline{y_i}$. Also we collect the constant temperatures of the wall segments into the parameter vector $\underline{c_i}$, which has the subscripts *i* to signify that the parameters belong to the integral conditions. For a periodic steady state solution:

1. The initial and final values of the periodic variables must be identical.
2. The integration results from the integral conditions for the wall temperatures must be zero.

But the above conditions are not adequate to determine a unique periodic steady state solution because the conditions can be fulfilled for any total mass of gas inside the gas domain. To get a unique solution an additional condition that determines the total mass of gas in the gas domain is needed.

The condition does not need to be formulated explicitly in the total mass of gas. Often it is convenient to specify a mean pressure, p_{mean}, at a point inside the gas domain. This mean pressure can either be an integral mean pressure or it can be a geometric mean pressure, i.e. an average of the minimum and maximum pressures observed during the cycle. The latter definition can be more practical when comparing with experimental data because it can be easier to measure the minimum and maximum pressures than to measure the integral mean pressure.

In the following we say that we can compute a result variable y_s as a monotone function of the amount of gas in the gas domain. We denote the desired value of this result variable as $SCTV$. The acronym for $SCTV$ stands for Scaling Condition Target Value. The reason for this name will become obvious when we in chapter 4 discuss methods for finding periodic steady state solutions to the models.

We can hence say that for a unique periodic steady state solution the initial values $\underline{y_{p,0}}$ for the periodic variables and the parameters values $\underline{c_i}$ for the integral conditions must satisfy the following conditions:

$$\begin{aligned} \underline{0} &= \underline{y_{p,0}} - \underline{y_p}\left(t_0 + \Delta t; \underline{y_{p,0}}, \underline{c_i}\right) \\ \underline{0} &= \underline{y_i}\left(t_0 + \Delta t; \underline{y_{p,0}}, \underline{c_i}\right) \\ 0 &= SCTV - y_s\left(t_0 + \Delta t; \underline{y_{p,0}}, \underline{c_i}\right) \end{aligned} \tag{17}$$

where $\underline{y_p}\left(t_0 + \Delta t; \underline{y_{p,0}}, \underline{c_i}\right)$ denotes the values of $\underline{y_p}$ at time $t_0 + \Delta t$ given initial values $\underline{y_{p,0}}$ and parameters $\underline{c_i}$.

3.3. Implementation of the modelling approach

Section 3.2 above describes the theory of the modelling approach developed in this work. In practice, there is a long way to go from writing down the general equations until one has a simulation up and running. The following subsections describe different aspects of the method of implementation chosen in this work.

3.3.1. Method of implementation: An equation based MUSSIM model

In this work it was chosen to implement Stirling machine and pulse tube cooler models in the simulation software *MUSSIM (MUSSIM* is an acronym for *Multi-purpose software for simulation). MUSSIM* is an equation based general purpose simulation tool, which has been developed by this author, and it has its origin in a preparatory thesis project.

The *MUSSIM* software is a collection of subprograms that, when combined with a model file, can be compiled into an executable simulation program. The code is written in *Fortran 95*. Some high level algorithms have been parallelised using the Message Passing Interface (MPI*)* and central parts of the code have been parallelised with a finer granularity using OpenMP. Hence *MUSSIM* simulation programs can benefit from using multiple processors (or multiple processor cores) when the necessary hardware, libraries, and compiler technology are available.

A consequence of using the *MUSSIM* software for implementation of models is that models basically end up being a collection of declarations and equations contained in a single source file. This type of model has both advantages and disadvantages compared to other types of implementations, such as the object oriented implementation used in the *Sage* software by Gedeon (1994) which hides the equations from the user.

The primary advantage of an equation based model is that such a model is very transparent and adaptable. It is my experience that users of *Sage* often end up wanting to study machine specific details in ways that are not supported by the predefined components of *Sage*. With an equation based model users have the possibility to modify the equations to suit their special needs.

The equation based approach also has possible disadvantages:

- Making the equation system requires insight and can be laborious and time consuming. This problem can be eliminated if the model equation system is generated automatically from user input about which components the model must contain and how they are to be connected. With such a model generator the user will have the choice to look at and modify the equation system if needed, or

to leave it be if the predefined components support all the needed functionality. Such a model generator could have a graphical user interface.

- The user can make changes that break a model, i.e. that cause the equation system to not correctly mimic the physics being modelled, or that simply make the equation system unsolvable. Test values, such as checksums for mass and energy conservation, can aid the user in making sure that none of the changes made to the equation system breaks the model.
- Model files can grow to be quite large (in practice on the order of a few thousand lines). Each component has its own block of equations even if the model contains multiple identical components. This may ultimately limit the number of components which can practically be combined in a single model. In my experience, however, this has not been an issue.

3.3.2. The structure of the MUSSIM software

The basic software structure of a *MUSSIM* simulation program is illustrated in Fig. 9. A main program can invoke high level numerical subprograms for solving IVPs, BVPs, algebraic equation systems (AEs), performing parameter studies, and performing optimisation. These numerical subprograms need to interact with the user defined model in order to calculate residuals, right hand sides, Jacobians, and so on. All interactions between the numerical subprograms and the model are performed through an optimising equation system interface (OESI). The OESI is designed to analyse the equation system at program start up. The analysis produces information that is used during execution of the program to minimise the computational effort needed to supply data to the numerical subprograms.

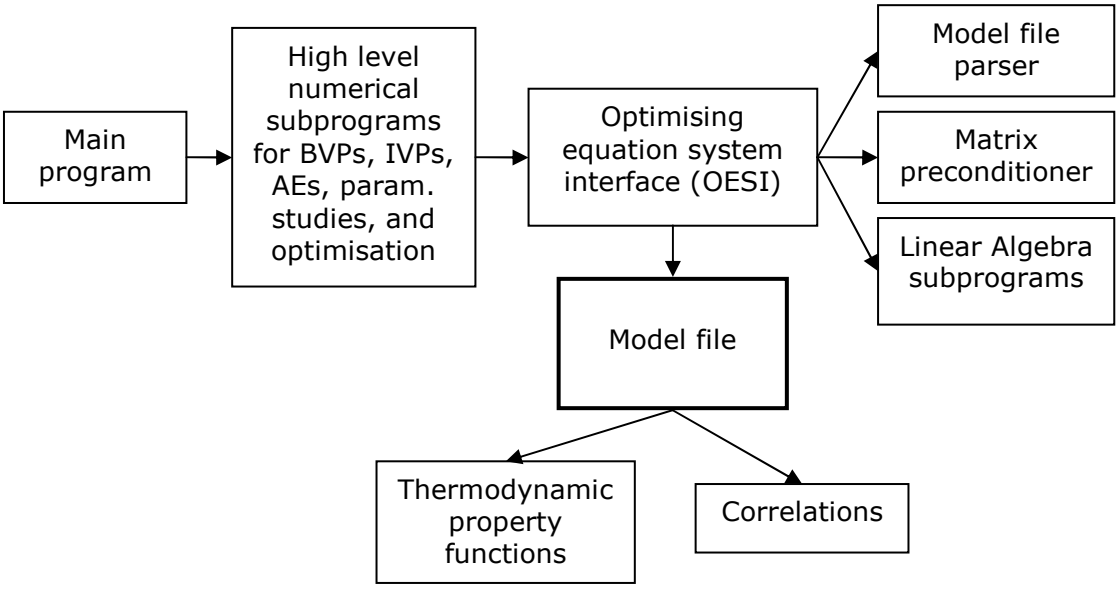


Fig. 9. Basic software structure of a *MUSSIM* simulation program.

Note that interaction with linear algebra subprograms is also performed through the OESI. This approach has been chosen to make it easier to switch between different matrix storage schemes and solvers. Changing to a new matrix storage scheme and/or solver only causes a few changes to the OESI and no changes to the high level numerical subprograms.

3.3.3. The mathematical structure of MUSSIM models

The *MUSSIM* software divides variables and parameters into the following main categories:

- ***Parameters*:** These are values that do not change during the integration of an IVP. Parameters can be both strictly input parameters or be parameters that must be determined from the simulation. An input parameter could be a tube diameter. A parameter, whose value must be determined from a simulation, could be a constant wall temperature.
- ***Static variables*:** These are variables that are determined from algebraic equations, i.e. variables that exhibit no "inertia".
- ***Dynamic variables*:** These are variables that are determined from differential equations.
- ***Shared evaluations*:** These are values that need to be calculated in order to evaluate the right hand sides of the algebraic equations and differential equations. In practice, the shared evaluations come from splitting up the complex right hand sides of equations into smaller, well defined calculations. When complex right hand sides are split up in this way, it will often become apparent that different right hand sides share common calculations. For instance, two neighbouring control volumes could share the calculation of the mass flow between them. The splitting allows these shared evaluations to be performed only once for each evaluation of all the right hand sides. It usually also turns out that many of these shared evaluations are independent and can be performed in parallel when using a multiprocessor computer. With proper (automated) optimisations the approach with the shared evaluations also has significant advantages when calculating Jacobians for the equation system using numerical differencing; this is explained in a section below. To get the maximum benefit of these optimisations it is a design objective, when implementing a *MUSSIM* model, that a large part of the computations needed to evaluate the right hand sides should be placed in the shared evaluations.

IVPs and BVPs that are to be solved using the *MUSSIM* software must be formulated as differential algebraic systems of equations (DAEs) of the form:

$$\begin{aligned} \frac{\underline{dy}}{\underline{dt}} &= \underline{g}\left(\underline{x}, \underline{y}, \underline{c}, \underline{s}, t\right) \\ \underline{0} &= \underline{f}\left(\underline{x}, \underline{y}, \underline{c}, \underline{s}, t\right) \end{aligned} \tag{18}$$

where $\underline{x}$ are static variables, $\underline{y}$ are dynamic variables, $\underline{c}$ are parameters, $\underline{s}$ are states, $\frac{\underline{dy}}{\underline{dt}}$ are the derivatives of the dynamic variables and where $\underline{f}$ and $\underline{g}$ are vector functions. In practice, equation systems in *MUSSIM* models also contain the above mentioned shared evaluations, $\underline{SE}$, as an intermediate stage in the calculations of $\underline{f}$ and $\underline{g}$:

$$\begin{aligned} \underline{SE} &= \underline{h}\left(\underline{x}, \underline{y}, \underline{c}, \underline{s}, t\right) \\ \frac{\underline{dy}}{\underline{dt}} &= \underline{g}\left(\underline{SE}, \underline{x}, \underline{y}, \underline{c}, \underline{s}, t\right) \\ \underline{0} &= \underline{f}\left(\underline{SE}, \underline{x}, \underline{y}, \underline{c}, \underline{s}, t\right) \end{aligned} \tag{19}$$

States are used for handling *events* where the equation system can undergo instant changes, i.e. it can shift from one state to another state. A change of states is essentially a discontinuity that occurs when some condition is fulfilled. A condition for an event could be defined as: *If P1 < P2 then open the Valve 1*. The equation system could then change from a state where *Valve 1* was closed to a state where *Valve 1* was open.

By manually specifying conditions for events it becomes possible to determine with high precision the times when events occur. Explicitly handling events, instead of just letting an IVP method with a variable step size algorithm stumble across the associated discontinuities, not only enhances the precision of the integration results but can also save computational effort.

The individual parameters, variables, shared evaluations, and equations in a *MUSSIM* model can be either in scalar or vector form. This is helpful when making models that involve spatial discretisation. The equations for a discretised component can be written in vector form so that refining the discretisation of the component amounts to changing a single parameter for the numbers of elements in the vectors.

Models created using the modelling approach presented above usually have equation systems which contain no static variables.

3.3.4. General implementation of MUSSIM models

A *MUSSIM* model is implemented in a model file. The model file is a Fortran 95 source file that contains a number of modules and subprograms (A module in Fortran 95 is basically a program unit with its own scope that can contain both declarations and sub programs. Other program units can selectively use variables and subprograms from a module). A *MUSSIM* model file contains the following main modules and subprograms:

- **The module *Model_info_module*:** This module contains information about the model source file and the input files that are associated with the model.
- **The module *User_variables_module***: This module is used for declaring variables to the Fortran compiler. Here it could be declared that, for instance, *WallTemperature_h1* is an array of 12 floating point values of a given kind. Typically the kind would specify 8 bytes pr. floating point value (double precision), but the program can also be compiled to use other number representations such as 4 byte (single precision) or 16 byte (quadruple precision) floating point values.
- **The subroutine *UserDeclarations*:** This subroutine is used for declaring variables, parameters, and so on to the OESI. Here it could be declared, for instance, that the array *WallTemperature_h1* from User_variables_module is an array of parameters.
- **The subroutine *ExecuteBeforeSimulation*:** This subroutine is used for computations that need only be performed once for every simulation. Computing geometry from input parameters and setting up the computational mesh is typically done here.
- **The subroutine *ExecuteBeforeResiduals*:** This subroutine is used during simulations to perform calculations that depend only on time.
- **The subroutine *SharedEvaluations*:** This subroutine is used for evaluating individual shared evaluations.

- **The function *ResidualsOfStaticEquations*:** This function is used for calculating individual residuals of algebraic equations.
- **The function *DerivativesOfDynamicVariables*:** This function is used for calculating individual right hand sides of differential equations.
- **The subroutine *ExecuteOnSolution*:** This subroutine is called after each step of the IVP solver. It can be used, for instance, for generating detailed output during a cycle.
- **The subroutine *ExecuteOnSample*:** This subroutine is called at each sample point, when an IVP solver has been instructed to find solution points with a fixed time interval between them. Sampling is typically used for generating output data with uniform time spacing between the values. But it can also be used for changing the values of parameters or variables with a fixed time interval. This is useful when modelling systems where controllers interact with the system at fixed time intervals.
- **The subroutine *ExecuteAfterSimulation*:** This subroutine is called when an IVP solver has finished an integration, for instance after each simulation of a complete cycle for a machine.
- **The subroutine *StateShifting*:** This subroutine is used for handling events (state changes) in a simulation.

3.3.5. A component oriented implementation of the equation based models

When using the modelling approach developed here for modelling Stirling or similar machines it can be advantageous to group the equations by components. By doing so it becomes easier to add or remove components in a model.

Connecting components

The fact that a staggered mesh is used in the model formulation causes a slight complication when components are connected.

When components are discretised into control volumes it appears natural to discretise them so that the whole interior of each component is divided into an integer number of solid line control volumes for mass and energy balances. At each intersection where two components are joined, one control volume for the momentum balance will then be shared between the two components. This is illustrated in Fig. 10. If components are to be connected in this way then one must choose a general scheme to decide which component should contain the equation for the velocity in the shared control volume.

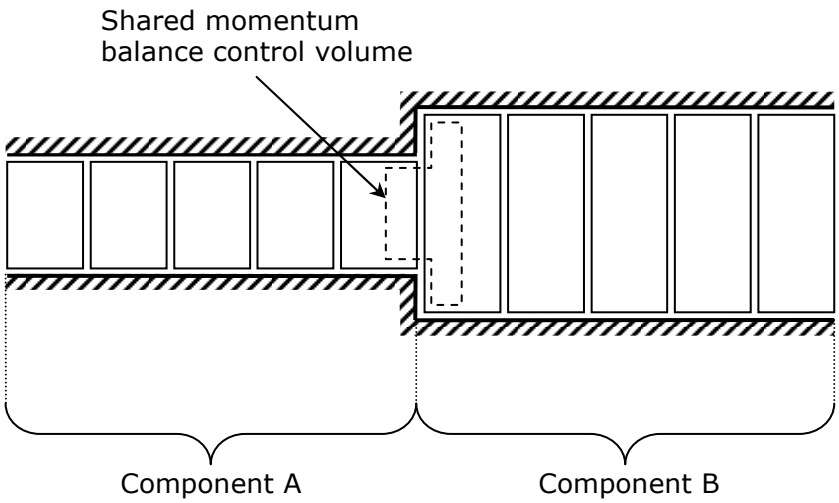


Fig. 10. Shared momentum balance control volume at connection between two components.

In this work an alternative approach, which is illustrated by Fig. 11, has been chosen. In the alternative approach there is no direct connection between components that have equations for the velocities at their outer boundaries. Instead these components can only be connected to special "manifold" components that do not have equations for the velocities at their boundaries.

In this work components such as cylinder volumes, heat exchangers, and regenerators have been chosen to have velocities at their outer boundaries. A single generic manifold component, which has only a single solid line control volume for mass and energy balances, has been used to connect the components. When there has been no distinct connecting volume between two components in the actual physical device being modelled, then the manifold component has been given the same properties as one of the neighbouring components, i.e. it has been used to model a short length of one of the neighbouring components.

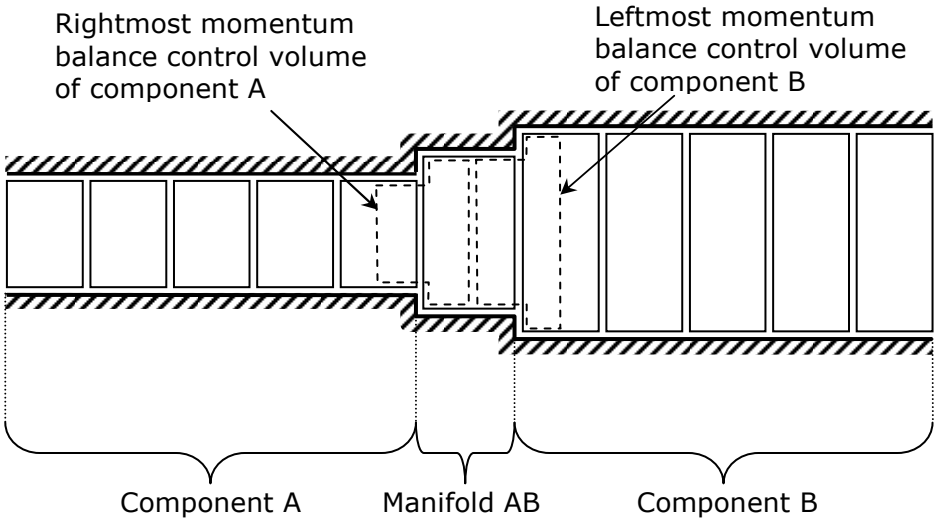


Fig. 11. Components connected through a manifold. Momentum balance control volumes are not shared directly between the components, only between the components and the manifold.

The manifold components are usually used to model volumes which are moderate in size compared to the amount of gas travelling back and forth during a cycle (see for instance the manifold volumes of the SM5 engine in Fig. 13 on page 52 below). The gas temperatures in the manifold volumes will hence tend to quickly approach the outlet temperatures of their closest upstream components.

As mentioned in section 3.2.6 extrapolation from upstream conditions are used for finding enthalpies at the intersections between components. In the single volume manifold components this extrapolation reduces to the upstream approximation. The manifolds thus function as mixing zones between components. During flow reversals the manifold components have a smoothing effect and they eliminate the step inputs in temperature to the spatially discretised components that could otherwise occur.

This method for connecting components corresponds to assuming that the gas is perfectly mixed inside the manifold volumes. The assumed perfect mixing, which smoothes discontinuities and hence has significant numerical benefits, is a crude approximation to reality. But capturing the true amount of turbulent mixing in manifolds is beyond a one-dimensional model anyway, and assuming no mixing at all does not appear to be any more justifiable than the assumption of perfect mixing.

Organisation of computations

The computations for the different components have been implemented in the *MUSSIM* models with the following general code organisation:

- *ExecuteBeforeSimulation* is used for performing initial geometry calculations from input parameters and setting up the computational mesh.
- *SharedEvaluations* is used for calculating all the terms that must be inserted in the governing equations. *SharedEvaluations* is hence where the bulk of the CPU time is spent when derivatives, residuals, and Jacobians are calculated.
- *DerivativesOfDynamicVariables* is used for calculating the derivatives corresponding to the governing equations from the values computed in *SharedEvaluations.*
- *ExecuteAfterSimulation* is used for computing heat conduction between control masses where constant wall temperatures have been assumed.

The shared evaluations for the different types of components can be organised into layers, so that there are no dependencies between the shared evaluations in each layer. The ordering into layers is performed automatically by the OESI and the shared evaluations can be entered into models in any order. The ordering of shared evaluations into layers is global for the entire model, i.e. it is not just performed for individual components. Hence components may cause the shared evaluations in other components to be split into more layers than a quick look would otherwise suggest, because there can be dependencies between the shared evaluations in different components.

If no other components influence the global ordering into layers, then the shared evaluations for the manifold components could be ordered into the following layers by the OESI:

1. Layer
   - Compute specific volumes, v, from the transformed variables in the governing equations (the transformed variables could, for example, be $\rho \cdot e$, e, and $\overline{V}$).
   - Compute specific internal energies, u, from the variables in the governing equations.

2. Layer
- Compute masses in control volumes from v and geometry.
- Compute temperatures, T, from v and u.

3. Layer
- Compute pressures, p, from T and v.

4. Layer
- Compute enthalpies, h_{gas}, from u, p, and v.

5. Layer
- Compute mass flow rates, $\dot{m}$, centrally in the solid line control volume from the mass flow rates at the outer boundaries (these are computed in the adjoining components).

6. Layer
- Compute flow velocities centrally in the solid line control volume from $\dot{m}$, v, and geometry.

7. Layer
- Compute heat exchange between gas and walls from flow velocity, gas properties, and geometry.

Under similar assumptions about the global ordering of the layers the shared evaluations for another component, a tubular heat exchanger, may be organised into layers in the following way:

1. Layer
- Compute specific volumes, v_i, from the variables in the governing equations.
- Compute velocities centrally in dashed line control volumes), $\bar{V}_j$, from the variables in the governing equations.

2. Layer
- Compute masses in control volumes from v_i and geometry.
- Compute velocities, $\bar{V}_i$, in solid line control volumes from $\bar{V}_j$ and geometry.

3. Layer
- Compute specific internal energies, u_i, from the variables in the governing equations and $\bar{V}_i$.
- Compute mass flow rates, $\dot{m}_i$, centrally in the solid line control volumes from $\bar{V}_i$, v_i, and geometry

4. Layer
- Compute temperatures, T_i, from v_i and u_i.
- Compute artificial dissipation from $\bar{V}_j$ and geometry.

5. Layer
- Compute pressures, p_i, from T_i and v_i.

6. Layer
- Compute enthalpies, $h_{gas,i}$, from u_i, p_i, and v_i.
- Compute pressures, p_j at centres of dashed line control volumes from p_i, the pressures in the adjoining manifolds, and geometry.
- Compute heat exchange between gas and walls in solid line control volumes from $\bar{V}_i$, gas properties, and geometry.

7. Layer
- Compute enthalpies, $h_{gas,j}$, centrally in dashed line control volumes from $h_{gas,i}$, enthalpies in the adjoining manifold volumes, and geometry.

8. Layer
- Compute temperatures, T_j, and specific volumes, v_j, from h_j and p_j.

9. Layer
- Compute mass flow rates, $\dot{m}_j$ centrally in dashed line control volumes from $\bar{V}_j$, v_j, and geometry.
- Compute flow friction, F_{wall}, in dashed line control volumes from $\bar{V}_j$, gas properties, and geometry.

10. Layer
- Compute rates of energy transport by advection, axial conduction, and turbulent axial conduction enhancement over boundaries of solid line

control volumes from $\dot{m}_j$, $h_{gas,j}$, $\bar{V}_j$ (kinetic energy), gas properties, and the temperature gradient between the adjacent solid line control volumes.

Other types of components contain additional shared evaluations. Regenerators, for instance, also compute the heat exchange between the gas and the regenerator matrix, axial heat conduction in the matrix, radial heat conduction between the matrix and the surrounding walls, and so on.

Note that it is only the first layer of the shared evaluations that uses the variables defined by the governing equations. The bulk of the shared evaluations do not depend on the transformations applied to the mass, energy, and momentum balance equations. It is possible to define the calculations in the first layer of shared evaluations conditionally so that they are valid for more than one transformation. By doing so one can make a model where one can change between different transformations by changing the value of an input parameter to the model.

3.3.6. The optimising equation system interface

By carefully placing executable statements in the correct subprograms in a *MUSSIM* model one can help to reduce the number of times that statements are executed during simulations. The OESI performs further optimisations to reduce the computational efforts needed to produce data for the numerical subprograms of the *MUSSIM* software. This section describes the sequence of steps in the optimisations performed by the OESI.

The optimisation process for the equation system contains the following phases:

1. **Parsing**: In this phase a partial Fortran 95 parser processes the model source file. The parser, which was written as part of this work, determines which variables and shared evaluations that each shared evaluation and equation depends on. Note that the parser is not used for building a new model from the model source file. The Fortran 95 source file is compiled as is and the executable code is optimised by the compiler. The parser is only used to determine the dependencies in the model so that the OESI can perform additional optimisations on the fly, in addition to the optimisations performed by the compiler.
2. **Mapping shared evaluations into layers:** During this phase the shared evaluations are divided into groups of independent shared evaluations. The first group (layer) contains the shared evaluations that do not depend on any other shared evaluations, i.e. the shared evaluations in the first layer depend only on static variables, and/or dynamic variables, and/or time. The shared evaluations in each following layer can depend only on variables, time, and/or on shared evaluations from the preceding layers. The shared evaluations inside each layer are hence independent of each other and can be calculated concurrently using multi-threading. The automated mapping of layers will reveal if there are any circular dependencies between the shared evaluations, i.e. if any of the shared evaluations need to be converted into algebraic equations.
3. **Accumulate the dependencies of the equations:** In this phase the non-zero pattern is mapped for the iteration matrix which is needed by the IVP solver. The dependencies of equations on shared evaluations are traced back through the

tree of dependencies for the shared evaluations, in order to determine all the dependencies of equations on dynamic variables and static variables.

4. **Optimise the iteration matrix non-zero pattern:** The next step is optimising the non-zero pattern of the iteration matrix. The optimisation is performed in order to minimise the fill in, i.e. the increase in the number of non-zero elements, during LU-decomposition of the iteration matrix. This optimisation is performed only at program start-up because all possible non-zero entries in the iteration matrix are known from the previous phases of the equation system optimisation process. During simulations some of the mapped elements may actually be zero. In this work the column reordering code *COLAMD* (University of Florida, 2005) has been used with good results for optimising the non-zero pattern.
5. **Generate info for optimised evaluation of Jacobians:** The next step is mapping the calculations that are needed to compute a Jacobian matrix by numerical differencing. To compute a column in a Jacobian matrix by numerical differencing one must perturb the value of a variable, re-compute the residuals, and compute the corresponding entries in the Jacobian by dividing the changes in the residuals by the perturbation applied to the variable. The mapping carried out in this phase ensures that only the individual shared evaluations and residuals, which actually depend on the perturbed variable, are updated.

A central aspect of compiler technology, that can significantly influence the computation times for the adopted method for making an optimisable equation system, is the ability to select individual shared evaluations and right hand sides to evaluate in a model. It is optimal to use a selection structure for the individual shared evaluations and right hand sides, which incurs a minimal overhead when selecting the blocks of code to execute. The *case* construct of Fortran is one construct that is superior in this respect to, say, successive if statements. The following is an example of a case construct:

```
integer :: i

integer, parameter :: N_block_1 = 1
                      N_block_2 = 10

integer, parameter :: block_1_offset = 1, &
                      block_2_offset = block_1_offset + N_block_1, &
                      block_3_offset = block_2_offset + N_block_2

⋮ ! Omitted code

select case(i)
  case(block_1_offset) ! Selector for i=1
    Code block 1
  case(block_2_offset:block_3_offset-1) ! Selector for i=2,3,4,...,11
    Code block 2
  case(block_3_offset) ! Selector for i=12
    Code block 3
  ⋮ ! Omitted code
end select
```

The case construct needs only one expression to be evaluated for each selection and this single evaluation determines which code block is executed. In the above example the evaluation of the single expression reduces to reading the value of the scalar integer *i*.

The values in the individual selectors must be constants known at compile time and this allows compilers to make powerful optimisations.

In this work the time spent selecting expressions to evaluate has been studied by looking at run time profiles for executable built with different compilers. The time spend selecting expressions can range from completely insignificant, when a highly optimising compiler is used, to a noticeable fraction of the running time, when the selection is less well optimised.

The main disadvantage of using the case construct is that it is necessary to recompile a model if the discretisation is changed, say by changing the value of a parameter like *N_block_2* in the example above, because this will change the values of the selectors.

One can imagine a model with only non-zero entries in the Jacobian. For such a model one would actually be wasting time on selecting individual right hand sides for evaluation when computing a Jacobian matrix because all the right hand sides would have to be computed anyway. If the residuals and right hand sides were also very cheap to evaluate then the overhead of using this implementation, instead of an implementation without selection, could be noticeable when computing Jacobians. For the application considered in this work, however, Jacobians are generally quite sparse and evaluating the right hand sides is computationally expensive

The selections in the approach described here also incurs an overhead when computing residuals. But the overhead, on good compilers, is insignificant compared to the time spent on computations in the right hand sides. Also the approach chosen here enables parallel (multi-threaded) evaluation of the right hand sides, and this advantage usually outweighs the minor overhead of the selections.

3.3.7. The MUSSIM Initial Value Problem Solver

The IVP solver of the *MUSSIM* software is a general Runge-Kutta type solver which can use both explicit and semi-implicit Runge-Kutta schemes. The schemes must have an embedded Runge-Kutta pair for error estimation and have a continuous extension which can be used when determining the exact times of events. The solver contains a small collection of both explicit and semi-implicit schemes and it is easy to add new schemes.

Explicit and semi-implicit Runge-Kutta schemes

The computational cost of performing an integration with a Runge-Kutta method depends both on the number of steps needed for the integration and on the average computational cost pr. step. The needed number of steps depends both on the desired accuracy, the order of the scheme, and on the properties of the equation system.

The semi-implicit Runge-Kutta schemes can typically use larger step sizes than explicit schemes when stiff problems are integrated with moderate requirements for accuracy. But the semi-implicit schemes require the iterative solution of a non-linear equation system at the stages of each step and hence the computational cost pr. stage is larger for semi-implicit schemes than for explicit schemes. The cost pr. iterative solution of the non-linear equation system in the stages of a semi-implicit scheme depends both on the properties of the equation system, on the needed accuracy, and on the step size; when the step size is large it is generally more difficult to compute an accurate guess for the solution at a new stage from previous solutions, and this can lead to a need for more iterations pr. solution.

The computational cost of integrating an equation system is thus influenced by many factors. For a new type of problem it is, hence, generally necessary to test the performance of different schemes at different tolerances in order to determine which scheme is optimal for the problem at hand. The differences in performance between different Runge-Kutta schemes can be significant and hence it is advantageous with respect to performing fast simulations that the IVP solver supports multiple schemes.

Recycling of Jacobians and iteration matrices for semi implicit schemes
The non-linear equation system in the stages of semi-implicit Runge-Kutta methods is solved using the AE solver of the *MUSSIM* software. To compute the stage solutions the AE solver requires an iteration matrix, $\underline{\underline{IM}}$, which is constructed from the Jacobian matrix for the equation system. The iteration matrix is used in LU-decomposed form during the solution process. For an equation system consisting of only ODEs the iteration matrix is given simply by:

$$\underline{\underline{IM}} = \underline{\underline{I}} - \Delta t_{step} \cdot \gamma \cdot \underline{\underline{J}} \tag{20}$$

where $\underline{\underline{I}}$ is the identity matrix, Δt_{step} is the step size, $\underline{\underline{J}}$ is the Jacobian matrix and γ is a constant specific to the semi-implicit Runge-Kutta scheme. If algebraic equations are also present then $\underline{\underline{IM}}$ will contain both rows corresponding to (20) and rows corresponding to the algebraic equations. In any case $\underline{\underline{IM}}$ depends both on Δt_{step} and on $\underline{\underline{J}}$.

The AE solver should only attempt to use an already LU-decomposed $\underline{\underline{IM}}$ from a previous solution to compute a new solution if the step size has not changed and if $\underline{\underline{J}}$ is still valid. The IVP solver of the *MUSSIM* software therefore passes information to the AE solver regarding changes in step size and regarding events, such as changes of states or samplings that influence the equation system, which invalidate $\underline{\underline{J}}$. This information enables the AE solver to minimise the computational efforts expended on maintaining an up to date version of the LU-decomposed $\underline{\underline{IM}}$.

Step size control
The step size controller of the IVP solver uses error estimates from embedded Runge-Kutta pairs in the Runge-Kutta schemes for estimating the errors pr. step. Prediction of step sizes is performed with a simple formula which depends only on the step size and the infinity norm of the estimated relative errors in the last step (Numerical Recipes, 1997):

$$\Delta t_{step,new} = \Delta t_{step,last} \cdot \left(\frac{safety\ factor \cdot tolerance}{\| last\ relative\ error\ estimates \|_{\infty}} \right)^{\frac{1}{Order\ of\ leading\ error\ estimator\ term}} \tag{21}$$

When semi-implicit schemes are used it is desirable to take several consecutive steps with the same step size because this makes it possible to recycle $\underline{\underline{IM}}$ between the steps if $\underline{\underline{J}}$ remains valid. The steps must be small enough to produce error estimates that stay within the required tolerance. But a small step size should not be maintained for too long if a significant increase in step size is possible because this could waste

computational effort. For semi-implicit schemes the step size controller therefore only increases the step size if the predicted possible increase in step size is larger than a threshold value. Also the step size is only decreased if the error, which the step size controller predicts will result from maintaining the current step size, is larger than a threshold fraction of the tolerance. There are also limiting values imposed on how large increases and decreases in step sizes are allowed.

The step size controller hence has a number of tuneable parameters. In this work reasonable values for these parameters have been determined by using the optimisation facilities of the *MUSSIM* software to minimise the time needed for short sample simulations.

More complex step size controllers, such as PI-controllers, also exist. But the limited experimentation performed in this work has not revealed any such controller which performs significantly better with semi-implicit methods than the simple expression in (21) for models built using the modelling approach described above.

3.4. A Stirling machine model for testing

This section describes a Stirling machine model that has been used for testing the modelling approach and the implementation described above. In most of the tests input parameters corresponding to the gas fired Stirling engine, SM5, by Carlsen (Carlsen and Bovin, 2001) have been used. But the model has also been used for a comparison between simulation results and experimental data for a commercially available free piston Stirling cooler by Twinbird (Twinbird, 2006).

3.4.1. The SM5 Stirling engine

The SM5 Stirling engine is a hermetically sealed unit and has a generator built into a sealed and pressurised crank case. The engine was designed for producing 9 kW of electric power but can deliver more than 10 kW electric power under optimal operating conditions. Two pictures of the SM5 engine and a drawing of the working volume of the engine are shown in Fig. 12.

The SM5 engine is a so called β-type Stirling engine. This means that, in contrast to the engine schematic shown in Fig. 2, the SM5 engine has only a single cylinder. The cylinder is divided into two cylinder volumes, the compression volume and the expansion volume, by an elongated displacer piston. The SM5 engine hence has the same five basic components as the schematic in Fig. 2: A compression volume, a cooler, a regenerator, an expansion volume, and a heater. But in addition to these five components the SM5 engine has an extra gas filled space which must be taken into consideration: The displacer clearance gap (a.k.a. the appendix gap). The appendix gap exists between the displacer piston and the cylinder wall.

The displacer separates the compression volume from the expansion volume, and hence it also separates the cold and hot ends of the engine. Any gas flow between the compression and expansion volumes, which travels through the appendix gap, is like a thermodynamic short circuit and has a negative impact on the performance of the engine. To minimise such gas flow there is a seal between the displacer piston and the cylinder wall. This seal must be kept cool to have a good wear resistance and hence it must be located near the compression space. This means that the appendix gap is open to the expansion volume at one end and extends all the way down to near the

compression space. There are practical limits to how small this gap can/should be made. In the SM5 engine the hot end of the displacer piston will grow on the order of magnitude 0.5 % in diameter when heated from room temperature to typical operating conditions. If the gap is made too small then the risk of jamming the displacer piston in the cylinder during heating or cooling of the engine becomes too great.

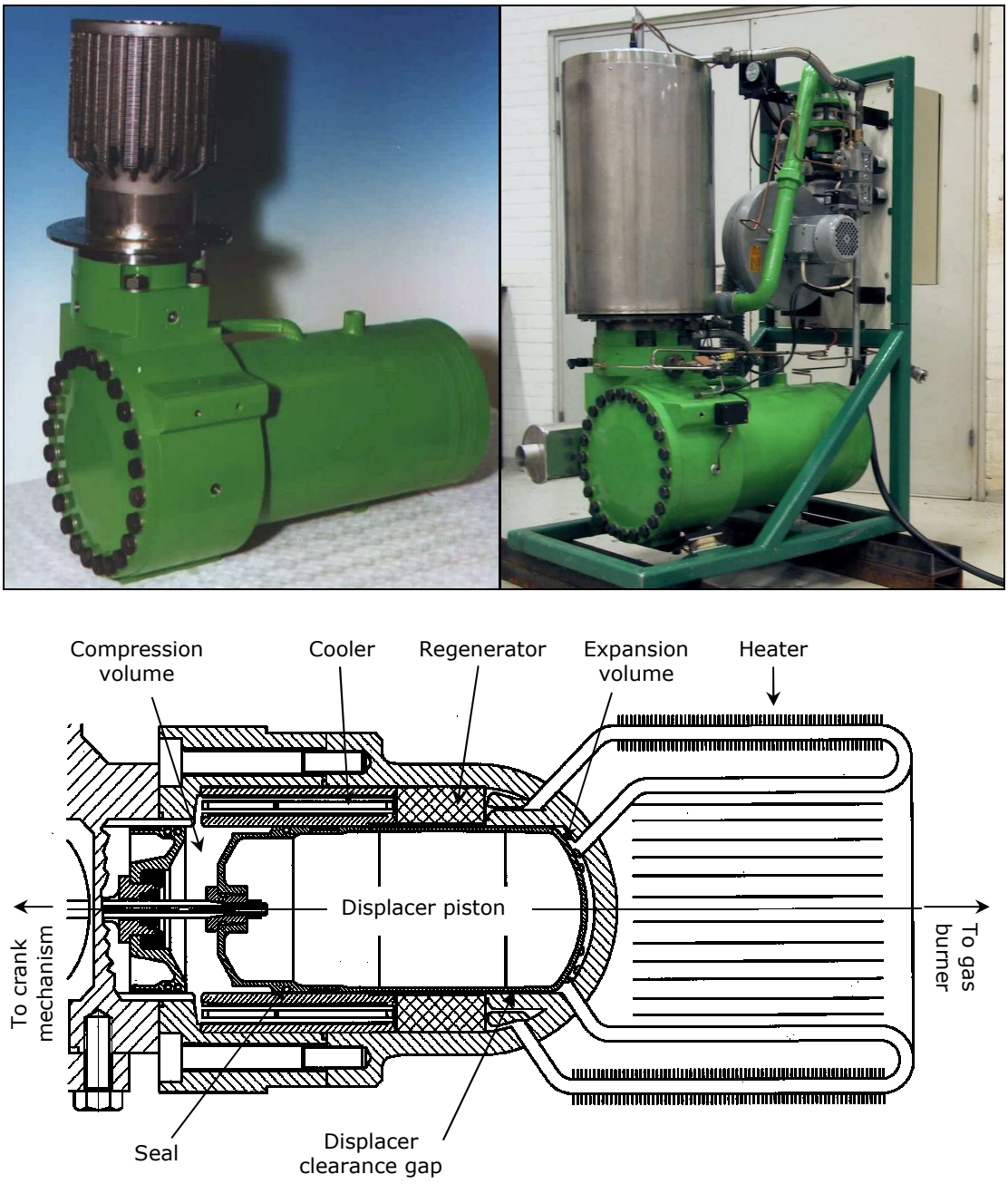


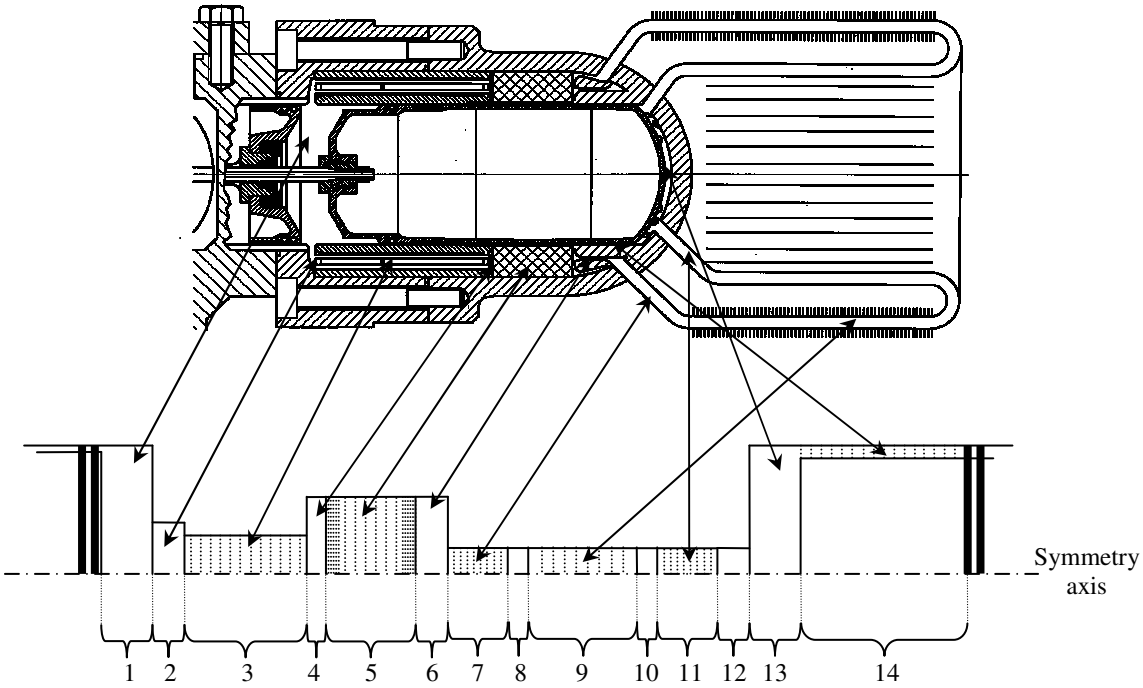
Fig. 12. The gas fired 9 kW Stirling engine, SM5, with its heater exposed (top left) and mounted in a frame with a natural gas burner system covering the heater (top right). Bottom: A drawing of the working volume of the engine.

3.4.2. The Stirling machine model

The Stirling machine model contains the five basic components of a Stirling machine and a component representing the appendix gap. The heat absorber is subdivided into

three discretised components in order to make it easier to take into account the variations in the outer boundary conditions of the heater; the tube sections closest to the expansion volume and the regenerator are covered by insulation while the central section of the tubes are exposed to radiation and hot flue gasses from the burner.

A schematic of the discretisation of the gas domain is shown in Fig. 13. The figure also illustrates the abstraction from the actual geometry of the SM5 engine to the one-dimensional discretisation of the gas domain in the model.



Regions of computational domain:

1:	Compression volume	7, 9, 11:	Heat absorber
2, 4, 6, 8, 10, 12:	Manifold volumes	13:	Expansion volume
3:	Heat rejector	14:	Displacer piston clearance gap
5:	Regenerator		

Fig. 13. Discretisation of the gas domain in the Stirling engine model. The abstraction between the components of the discretisation and the actual engine is shown by the arrows.

The solids in the engine are discretised axially as described in section 3.2.7. The temperatures of the regenerator matrix are modelled using the lumped formulation of Eq. (15). All other surface temperatures are assumed constant during each cycle. It has been verified that the lumped formulation is appropriate for the regenerator matrix control masses by simulating the radial temperature variations within single wires at different positions within the regenerator. Radial heat conduction is included between the cylinder wall and the components surrounding the cylinder (regions 2 through 6 in Fig. 13).

For the appendix gap there are some additional technicalities that must be considered when implementing the modelling approach described in section 3.2. These technicalities are due to the fact that the displacer is moving relative to the cylinder wall. The consequence of this is that the control volumes containing the gas in the gap slide over the discretisation of the cylinder wall, i.e. a sliding mesh is needed. The details of how the technicalities related to modelling the appendix gap have been addressed are described in the accompanying *Paper D*.

3.4.3. Initial values for testing the Stirling machine model as an initial value problem

When testing the model formulation it is useful to be able to directly compare simulation results for different spatial discretisations. There are, at least, two different options when choosing initial values for such studies:

1. Explicitly prescribing initial temperature-, pressure-, and velocity distributions that depend only on the axial space coordinate and hence are independent of the discretisation. With this approach one must usually expect to have a significant transient at the beginning of the cycle when the gas settles from the prescribed initial distributions and onto distributions that corresponds to the conditions in the machine.
2. Using the initial temperature-, pressure-, and velocity distributions from periodic steady state solutions. These distributions will only become independent of the discretisation in the sense that when the discretisation is sufficiently refined it should produce a grid independent solution. With this approach there should be no transient at the beginning of the cycle.

Results for both of these options are presented below.

The Stirling machine model has a subroutine for generating a standard initial guess for the distributions of temperatures-, pressures-, and velocities in the machine. In this initial guess all gas velocities are zero. The pressure is taken to be constant and equal to the cyclic mean pressure throughout the machine. The guess for the temperature distribution is generated from the input parameters to the model. The temperature distribution of the standard initial guess is illustrated in Fig. 14. The distribution corresponds to the input parameters, both the geometry and operating conditions, used in the tests below. For a given set of input parameters this temperature distribution is a function of the axial space coordinate only, i.e. it is independent of the discretisation.

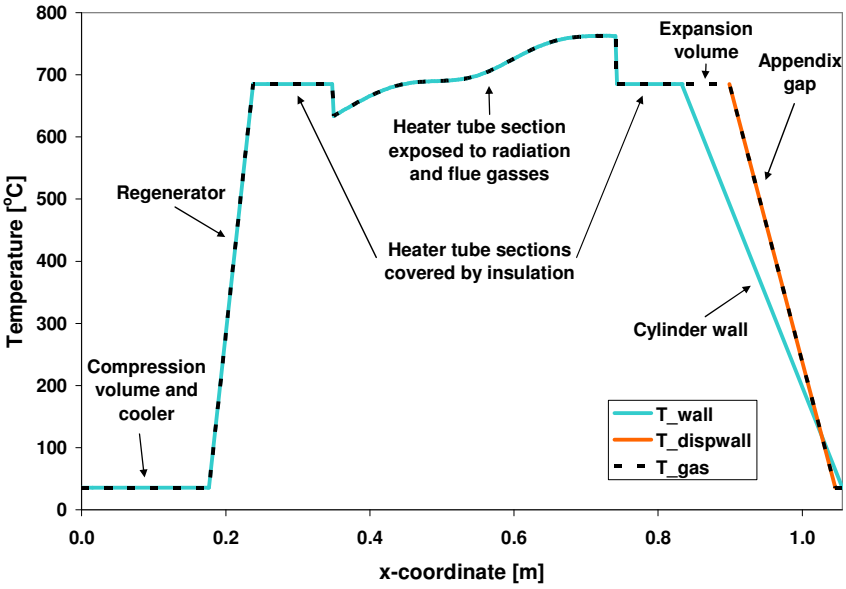


Fig. 14. Standard initial guesses for the temperature distributions in the gas, the stationary walls, and the wall of the displacer in the Stirling machine model. The axial positions of the components are laid out as in the bottom drawing in Fig. 13. The distributions correspond to the input parameters used for testing the modelling approach.

3.4.4. The spatial discretisations used for the testing

In the tests performed with the Stirling machine model the fineness of the spatial discretisation was varied. The discretisations of varying fineness were derived from a common base discretisation with six solid line control volumes pr. discretised component. The discretisation was refined by successively doubling the number of solid line control volumes pr. discretised components to, respectively, 12, 24, 48, 96, 192, and 384 control volumes.

The control masses on the cylinder wall cover a section of wall which is longer than the appendix gap. Hence it appeared natural to have a slightly larger number of control masses on the cylinder wall than there are solid line control volumes in the appendix gap. In the base discretisation, with six solid line control volumes in each discretised component, it was chosen to divide the cylinder wall into 10 control masses. The number of control masses on the cylinder wall was also doubled with every refinement to, respectively, 20, 40, 80, 160, 320, and 640.

As explained in the accompanying *Paper D* the sliding mesh in the appendix gap causes discontinuous derivatives in the heat fluxes to the cylinder wall control masses. These discontinuities are handled as events (changes of states). The number of events pr. simulated cycle increases with the number of control masses on the cylinder wall. The Jacobian matrix for the IVP method must be updated at each event. The refinement of the spatial discretisation therefore caused a need for more Jacobians, end thereby also for more LU-decompositions.

The numbers of control volumes, ODEs, and shared evaluations in the different refinements of the spatial discretisations are shown in Table 1 together with the numbers of events pr. simulated revolution of the SM5 engine.

Refinement	N_cv/comp	N_cm/cyllwall	N_cv	N_ODEs	N_SEs	N_events/cycle
0	6	10	44	214	952	16
1	12	20	80	380	1728	32
2	24	40	152	712	3280	62
3	48	80	296	1376	6384	120
4	96	160	584	2704	12592	236
5	192	320	1160	5360	25008	466
6	384	640	2312	10672	49840	934

Table 1: Solid line control volumes pr. component, control masses on cylinder wall, total number of solid line control volumes, number of ODEs, number of shared evaluations, and the number of events pr. cycle for the different refinements of the spatial discretisations of the Stirling machine model. The numbers of events pr. cycle are specific to the geometry of the SM5 engine.

It is possible to refine the discretisation further than this and still have memory requirements that are manageable on a standard PC. But the requirements for CPU time on a PC would make such fine discretisations impractical for simulations of more than a few cycles.

It should be emphasized that discretisations as fine as the last refinements in Table 1 usually have little relevance for practical applications; they are used here only to study the convergence properties of the modelling approach. Also it will often vary between different components how many control volumes are needed to obtain the desired accuracy. A significant reduction in the total number of control volumes in a model can result from only using fine discretisations in the components where this is necessary.

For most practical applications it would usually be a waste of effort to use a spatial discretisation that provides more than 2-3 significant correct digits for the power

output and the heat intake. By comparison the error due to the assumption of one-dimensional flow can be expected to be of the order of magnitude of several percent.

For many purposes discretisation errors up to several percent can be acceptable. One such case is when optimisation is performed. In this case one can first perform the optimisation using a coarse discretisation, and then use a finer discretisation only to compute how much the optimum is relocated by refining the discretisation.

3.4.5. An example solution for the SM5 engine

In some of the following tests the convergence of the spatial discretisation have been studied. In that context it is relevant to have an idea of how the solutions actually look. Fig. 15 shows an example of a how the pressure-, temperature- and velocity distributions in the gas domain can look in a periodic steady state solution for the SM5 engine.

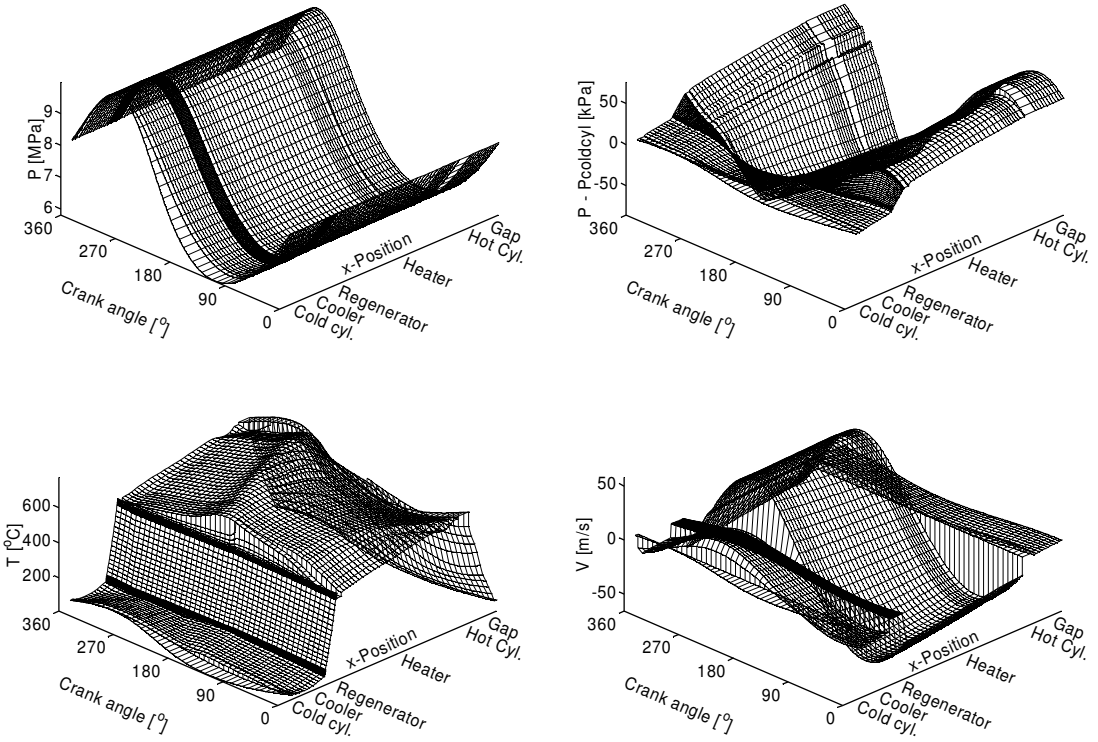


Fig. 15. Distributions in space and time (plotted as crank angle) of pressures, pressure losses, temperatures, and velocities in a periodic steady state solution for the SM5 engine.

In Fig. 15 the top left plot shows the variations in time of the pressure and the shape of the computational domain. But it is difficult to see any spatial pressure variations. The plot to the top right shows the difference between the pressures in the computational domain and the pressure in the compression volume. The pressure difference changed sign twice in the cycle as the gas flow between the cylinder volumes changed direction twice. It can be seen that the pressure gradients were larger in the regenerator than anywhere else in the engine.

The effects of inlet pressure losses are also visible in the plot of the pressure differences in Fig. 15. The clearest example is at the inlet from the expansion volume to the heater

at approximately 90 degrees crank angle. Here a pressure drop was caused by the acceleration of the gas from almost stationary in the expansion volume to the flow speed in the heater, and by an inlet pressure loss coefficient that was applied at the inlet. Wave phenomena are visible in the plot near the expansion volume in the last quarter of the cycle. They were induced when low Reynolds numbers during a flow reversal caused a momentary dip in the heat exchange predicted by the empirical correlations used in the model. Note that the visible waves are in the time wise direction; no waves can be seen travelling along the domain.

The bottom left plot in Fig. 15 shows the large variations in temperature in the gas domain caused by heat exchange and by the compression and expansion of the gas. Steep temperature gradients can be seen in the regenerator and in the appendix gap. The gas temperature oscillations in the regenerator are small due to the powerful heat exchange with the regenerator matrix and the large heat capacity of the matrix. Between the regenerator and the heater one can see how the connecting manifold volume smoothed what could have almost been a step input in temperature to the heater at flow reversal.

The bottom right plot in Fig. 15 shows the flow velocities in the domain. The velocities are plotted as positive when the flow was towards the expansion volume. Abrupt changes in velocities are clearly visible at the area discontinuities between the components. The largest velocities can be seen in the heater of the engine where it is just visible that velocity changes travelled along the domain; they did not occur simultaneously throughout the domain.

3.5. Overview of tests

A number of tests have been performed to illustrate the properties of the modelling approach and the implementation. A brief overview of these tests is presented here.

1. **Tests of the optimising equation system interface.** The purpose of these tests was to illustrate performance enhancing features of the optimising equation system interface of the *MUSSIM* software. It was studied how much the OESI can speed up the calculation of Jacobians. It was also quantified how much multi-threading can speed up simulations. The performance of three different linear algebra packages was also tested.
2. **Tests of the *MUSSIM* IVP solver.** The purpose of these tests was to demonstrate the accuracy and performance of the IVP solver of the *MUSSIM* software when it is applied to the Stirling machine model from section 3.4. The performance and accuracy of seven Runge-Kutta schemes were compared. The influence of the spatial discretisation on the IVP solvers accuracy for integral values, such as the work output and regenerator loss, was studied. The abilities of the IVP solver to minimise the needed number of Jacobian updates and LU-decompositions were illustrated.
3. **Tests of the spatial discretisation of the modelling approach.** These tests were carried out to test the convergence of the spatial discretisation of the modelling approach. The stability of the spatial discretisation was tested by continuously refining the spatial discretisation. The consistency of the discretisation was tested for two test problems with known analytical solutions. The conservativeness of the formulation was also documented.

4. **Validation of the modelling approach against experimental data and simulation results from *Sage* and *PROSA*.** These tests were performed to show that the modelling approach is sufficient to produce simulation results in good agreement with experimental data and with simulation results from current state of the art simulation software. The Stirling machine model from section 3.4 was validated against the SM5 Stirling engine and the Twinbird free piston Stirling cooler. A comparison of results from the Stirling machine model to *PROSA* simulation results for the Twinbird cooler was also performed. Results for two pulse tube cooler models based on the modelling approach presented above were also compared to simulation results for two pulse tube cooler models created using the *Sage* software.

The individual tests and the results are described in the sections that follow.

3.6. Tests of the optimising equation system interface

The optimising equation system interface (OESI) can help to enhance the performance of the IVP solver in the following ways:

- It can speed up the calculation of Jacobian matrices by reducing the number of operations needed to compute them.
- It can, assisted by the parser of the *MUSSIM* software, determine layers of independent shared evaluations, so that the shared evaluations within each layer can be calculated in parallel using multi-threading. The evaluation of the residuals themselves and the LU-decomposition of iteration matrices can also use multi-threading.
- It can use different linear algebra packages, so that one can choose the package that performs best for the problem at hand.

To illustrate these features the following studies and measurements have been performed:

1. The patterns of non-zero elements in the Jacobians corresponding to the 0^{th} and the 6^{th} refinements of the spatial discretisation from Table 1 have been visualised.
2. The speedups achieved by the OESI for calculating Jacobians have been measured for different spatial discretisations.
3. The speedups achieved by going from a single-threaded to a dual-threaded program have been measured for simulations of one unsteady cycle of the SM5 engine with different spatial discretisations.
4. The influence of using equations of state of different complexities for the gas in the SM5 engine on the simulation time and on the speedup from dual-threading has been measured.
5. The time needed for performing LU-decomposition of Jacobians corresponding to different spatial discretisations have been measured for three different linear algebra packages.

3.6.1. Optimised evaluation of Jacobians

The structure of the Jacobian matrix for the Stirling machine model
Fig. 16 shows the patterns of non-zeros in Jacobians for the Stirling machine model corresponding to the 0th and the 6th refinements of the spatial discretisation from Table 1. The patterns represent the maximum possible number of non-zeros. During simulations some of these elements may actually be zero valued.

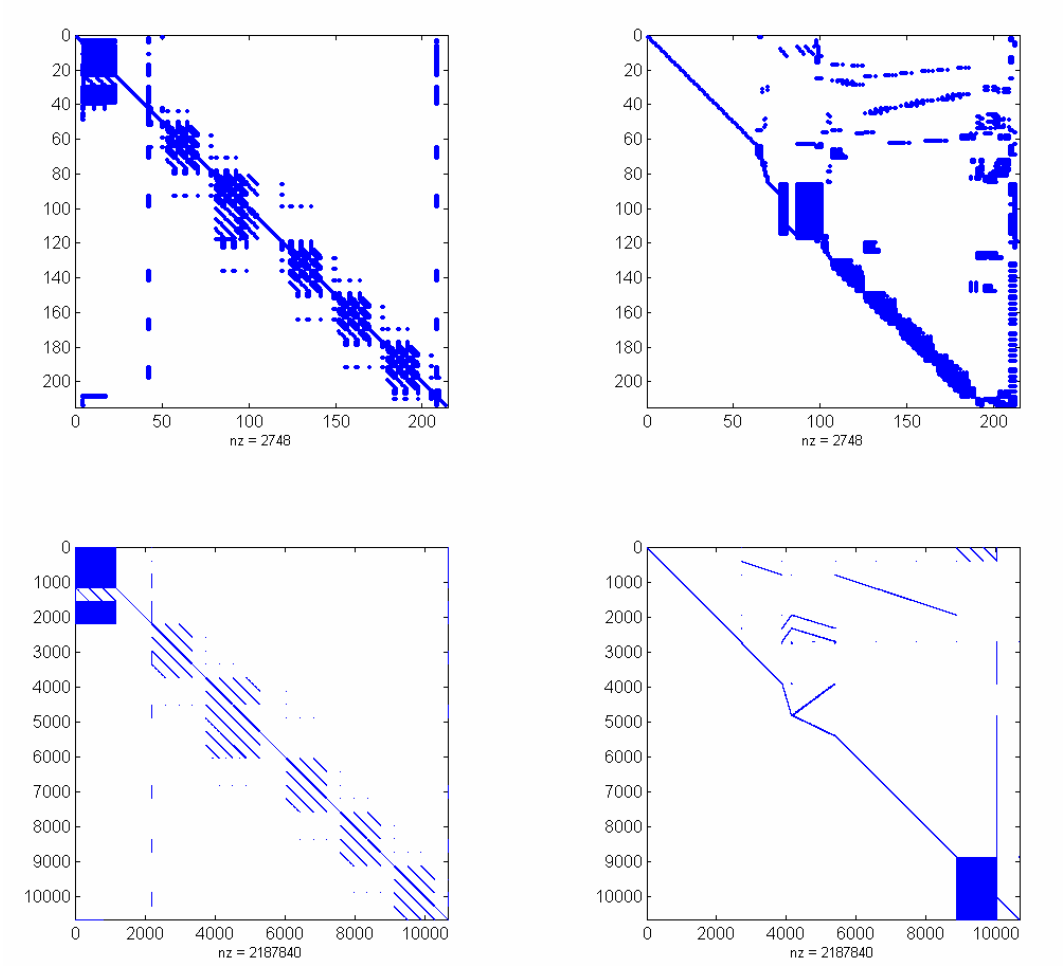


Fig. 16. Non-zero patterns for the 0th (top) and 6th refinements (bottom) of the spatial discretisation. The plots on the left show the non-zero patterns corresponding to the order in which variables and equations are set up in the Stirling machine model. The plots on the right show the same non-zero patterns optimised for minimum fill-in during sparse LU-decomposition.

In the plots on the left hand side in Fig. 16 one can clearly identify six clusters of non-zeros corresponding to the six spatially discretised components in the model. Counting from left to right the clusters represent: The appendix gap, the cooler, the regenerator, and the three sections of the heater.

The cluster corresponding to the regenerator is larger than the clusters for the cooler and heater, because this component has extra equations to model the regenerator matrix and both the inner and outer walls of the annular regenerator cavity.

The cluster of non-zeros corresponding to the appendix gap is very dense because of the moving mesh used in the gap. The parser cannot determine from the model source code how the control volumes in the gap will line up with the control masses on the cylinder wall. Also splines that extend the entire length of the gap are

used in order to obtain a smooth and conservative formulation for the gap. The technicalities due to the moving mesh are described in more detail in the accompanying *Paper D*.

Fig. 16 shows that the cluster of non-zeros due to the equations for the appendix gap begin to dominate the non-zero pattern of the Jacobian matrix when the spatial discretisation is refined. For the 6^{th} refinement approximately 95 % of the non-zero elements in the Jacobian come from the equations for the appendix gap. The average number of non-zeros pr. equation for the entire Jacobian increases by a factor of 16 from the 0^{th} to the 6^{th} refinement.

For the very fine discretisations of the appendix gap the computational performance of the Stirling machine model suffered due to the inclusion of the equations for the appendix gap into the model. In addition to increasing the time needed to compute Jacobians the appendix gap equations also increased the number of events, and hence the numbers of needed Jacobians and iteration matrices, and the computational cost of performing linear algebra on the iteration matrix.

For practical applications, however, adequate accuracy for the appendix gap can be achieved with relatively coarse discretisations. For production use of the Stirling machine model the added computational cost due to the inclusion of the appendix gap in the model has not been an issue.

Speedup for calculating Jacobians

The speedup from using the OESI for calculating Jacobians has been measured for simulations of one unsteady cycle of the SM5 engine for each of the spatial discretisations in Table 1.

Un-optimised evaluation of a Jacobian matrix for a system of *N* equations by one-sided numerical differencing usually requires *N+1* evaluations of all the residuals. For each simulation the speedup for computing Jacobians has therefore been calculated as the time needed to evaluate the residuals N+1 times for each Jacobian evaluation divided by the actual time needed by the OESI for computing Jacobians. The results are shown in Table 2.

Refinement	N_cv/comp	N_cm/cyllwall	N_ODEs	Speedup
0	6	10	214	31.2
1	12	20	380	29.8
2	24	40	712	52.0
3	48	80	1376	62.3
4	96	160	2704	57.8
5	192	320	5360	45.7
6	384	640	10672	31.9

Table 2. Measured speed achieved by the OESI for computing Jacobians for each of the spatial discretisations in Table 1.

Table 2 shows that speedup factors between 31 and 62 were achieved for computing Jacobians. The speedup initially increased with the size of the Jacobian. But the speedup decreased again when the discretisation was further refined. For the finest discretisations the many non-zero elements due to the moving mesh in the appendix gap dominated the Jacobian matrix. Evaluating these elements also became more expensive when the size of the moving mesh was increased. This was the reason why the speedup decreased for the finest discretisations.

3.6.2. Speed up of IVP solving from multi-threading

The speedup from multi-threading, or more precisely dual-threading, has been measured for the simulation of one unsteady cycle of the SM5 engine for the different spatial discretisations in Table 1. The tests were run a PC with an AMD Athlon64X2 3800+ processor operating at 2 GHz clock frequency, i.e. on a platform where two processor cores with independent caches (512 KB L2 cache pr. core) share the same memory bus bandwidth. The Intel Visual Fortran 9.1 compiler, which supports OpenMP, was used for compiling the executables used for these tests.

First the simulations were run in single-threaded configurations so that there was no overhead from the additional code needed for multi-threading. These runs were profiled to measure the fractions of the total running time spent on:

- Calculating shared evaluations for residual updates.
- Calculating the residuals.
- Performing LU-decompositions.

The sums of these three fractions were the total parallel (multi-threaded) fractions of the running times. The measurements were performed using the built in profiling tool of the *MUSSIM* software, which typically causes less than 2 % runtime overhead for the Stirling machine model.

The same simulations were then profiled in parallel configurations (two threads). The speedups were calculated as the times needed during the single-threaded runs divided by the times needed during the dual-threaded runs.

The gas in the engine was modelled as an ideal gas in the tests of the speedup from multi-threading. When more complex equations of state are used then the fraction of the running time spent doing shared evaluations increases. Using an equation of state with more computationally expensive property function evaluations hence influences the multi-threaded scalability of a model. To document this effect the measurements for the the first refinement of the spatial discretisation in Table 1 were repeated for two additional equations of state: The Redlich-Kwong equation of state for helium and a more complex equation of state by (Reynolds, 1979) for CO_2.

Profiles of single-threaded simulations

The profiles of the single-threaded simulations are shown in Table 3.

Ref.	N_ODEs	Norm. wall time	SE fraction	Deriv. fraction	LU fraction	Par. fraction
0	214	1.0	0.832	0.037	0.013	0.883
1	380	1.4	0.842	0.021	0.014	0.877
2	712	3.1	0.820	0.027	0.020	0.867
3	1376	7.7	0.758	0.023	0.059	0.841
4	2704	24.6	0.615	0.017	0.189	0.822
5	5360	135.2	0.291	0.009	0.552	0.852
6	10672	1134.4	0.091	0.002	0.834	0.927

Table 3. Profiles of single-threaded simulations of one unsteady cycle of the SM5 engine. Shown are the normalised running times, the fractions of the running time spent on shared evaluations for residual updates, on calculating the derivatives, and on performing LU decompositions. The total fractions of the running time used for tasks which have been parallelised by multi-threading are also shown.

Table 3 shows that the dependence of the running time on the fineness of the spatial discretisation was very non-linear. The increase in running time was quite moderate for

the first few refinements but increased rapidly for the last refinements. The dependence of the running time on the number of equations was not a simple power function.

For the coarsest discretisation 83 % of the running time was spent on calculating shared evaluations for derivative updates. When the discretisation was refined the profile gradually changed so that a smaller fraction of the running time was spent on shared evaluations and more time was spent on LU-decompositions. This was partly because the size of the Jacobian matrix increased with the refinements and partially because the refinements of the discretisation of the cylinder wall increased the number of events and, thereby, the number of LU-decompositions. For the finest discretisation 83 % of the running time was spent on LU-decompositions. State changes caused 93 % of the Jacobian updates and 60 % of the LU-decompositions.

The fractions of the running time spent calculating the derivatives were not very significant.

The total fractions of the running time spent in parallel sections varied between 82 % and 93 % for all the discretisations.

Speedups from dual-threading

Table 4 shows the measured speedups achieved by the dual-threaded runs.

Ref.	N_ODEs	SE speedup	Deriv. speedup	LU speedup	Speedup
0	214	1.62	1.02	1.21	1.45
1	380	1.69	0.85	0.95	1.54
2	712	1.69	1.31	0.98	1.55
3	1376	1.79	1.85	1.25	1.54
4	2704	1.84	1.57	1.72	1.57
5	5360	1.80	1.42	1.80	1.60
6	10672	1.85	1.13	1.66	1.54

Table 4: Measured speedups for the simulation of one unsteady cycle of the SM5 engine. Shown are the speedups for performing shared evaluations, for calculating derivatives, and for performing LU-decompositions. The resulting overall speedups for the entire simulations are also shown.

The speedups for the shared evaluations and the LU-decompositions generally improved when the discretisation was refined. Since these two tasks constituted the majority of the parallel fractions of the simulations the total speedups for the simulations generally also improved when the discretisation was refined. The total speedup factors for the simulations varied between 1.45 and 1.60.

The speedups for the calculation of the derivatives were more sporadic, but in any case this task constituted only a minor fraction of the total running time.

Imperfect load balancing, i.e. the inability of the code to distribute the computational load perfectly between the two processor cores, was suspected to be the main reason that the speedup factors for the individual parallel sections were well below 2.0. But shared memory programming is generally hard to analyse and many other factors, such as the two processor cores invalidating the contents of each others caches (a phenomenon known as false sharing) and the two cores competing for the same memory bandwidth, could also be significant.

Measurements of the speedup from multi-threading were also attempted on a Sun Fire 15000 server. The speed of inter-processor-core communication varied significantly on the Sun Fire 15000 because it consisted of multiple system boards with multiple dual-core processors on each system board. The measurements on the Sun Fire

15000, unlike the measurements on dual-core PC, resulted in sporadic results. This indicated that the overhead from inter-processor communication was significant.

Influence of the equation of state for the gas on the running time and the multi-threaded scalability
Table 5 shows the results for the simulations performed with different equations of state for the first refinement of the spatial discretisation from Table 1.

Gas:	Helium	Helium	CO_2
Equation of state:	Ideal gas	Redlich Kwong	Reynolds PTV3
Normalised seq. wall time	1.00	1.45	2.21
SE fraction	0.84	0.88	0.89
Derivativs fraction	0.02	0.02	0.01
LU decomp. fraction	0.01	0.01	0.02
Total parallel fraction	0.88	0.91	0.91
SE speedup	1.69	1.77	1.85
Derivatives speedup	0.85	1.61	1.02
Total speedup	1.53	1.64	1.67

Table 5. Speedups measured for the simulation of one unsteady cycle of the SM5 engine using the first refinement of the spatial discretisation and different equations of state for the gas in the engine.

The results in Table 5 show that the sequential running time increased with the complexity of the equation of state. The total parallel fraction, however, increased only slightly because the time spent on calculating Jacobians also increased.

The speedup for the shared evaluations, which contained all the property function evaluations, increased significantly with the complexity of the equation of state. The speedup for the shared evaluations increased the total speedup from 1.53 for ideal gas helium to 1.67 for the more complex equation of state for CO_2.

Summary of multi-threaded tests
The parallel fractions of the simulations accounted for 82-93 % of the sequential running time.

Speedup factors from 1.45 to 1.67 were achieved for equation systems of moderate size on a PC with a dual-core processor. Load balancing appeared to limit the speedup. In tests on a Sun Fire 15000 supercomputer the overhead from inter-processor-core communication appeared to also limit the speedup.

Given the fractions of the running time spent in parallel sections and the fact that load balancing appeared to be an issue it would appear, that the speedup achievable by this implementation for equation systems of such moderate size would probably be less than 5 even on a supercomputer with very fast inter-processor communication.

3.6.3. Comparison of Linear Algebra packages

The run time performance of three different linear algebra packages were tested for the Stirling machine model. In the tests it was measured how much time was needed to perform LU-decomposition of iteration matrices during simulations for the different refinements of the spatial discretisation. The packages included in the tests were:

1. The custom linear algebra package, DCRM (*Dynamic Compressed Row Matrix*), of the *MUSSIM* software.
2. The *PARDISO* package from the commercial Intel Math Kernel Library (commonly known as Intel MKL).
3. LAPACK from Intel MKL.

The packages were tested in both single-threaded and dual-threaded configurations on the PC with the dual-core AMD Athlon64X2 processor. The results from the tests are plotted in Fig. 17.

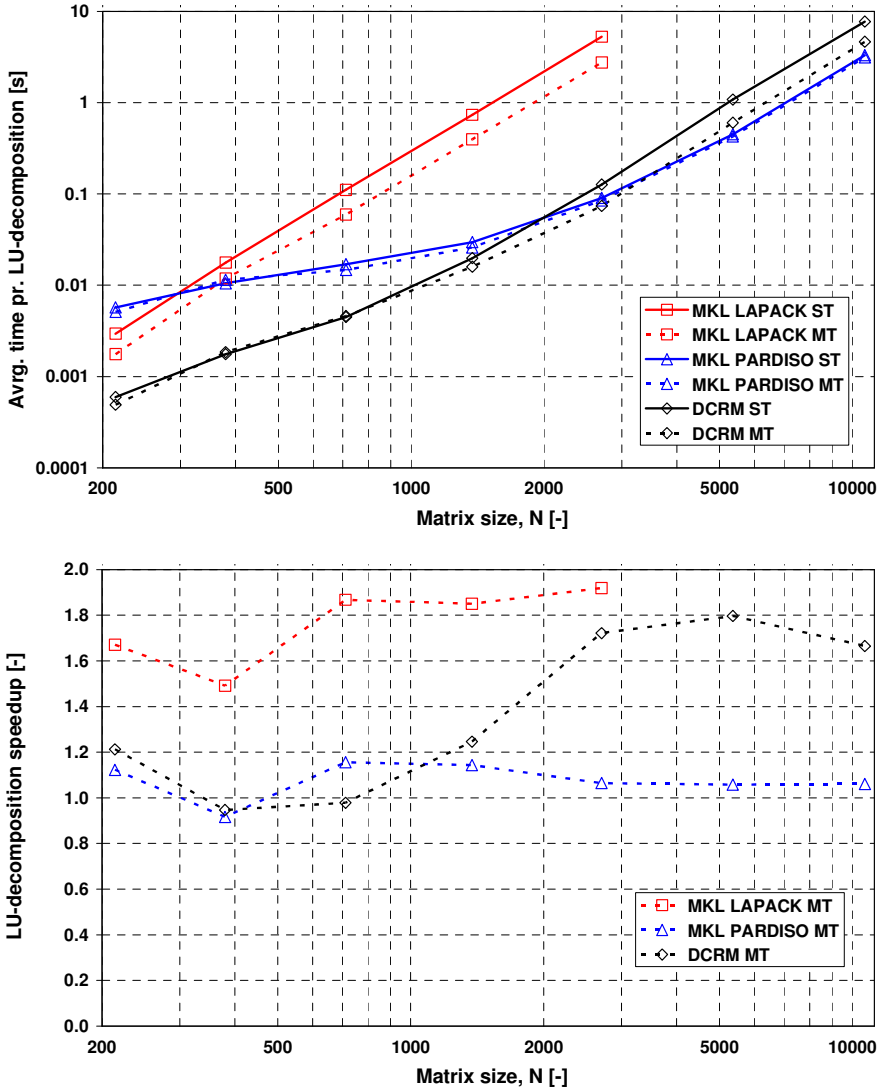


Fig. 17. Comparison of needed time pr. LU-decomposition of the iteration matrix for different discretisations (top) and speedup from dual-threading (bottom). Results are shown for LAPACK and PARDISO from the *Intel Math Kernel Library* (MKL) and for the built in DCRM package of the *MUSSIM* software.

Fig. 17 shows that the optimal choice of linear algebra package depended on the problem size. For matrix sizes below approximately 2000 x 2000 the DCRM package was the fastest of the tested packages. For matrix sizes above 4000 x 4000 PARDISO was the fastest package. LAPACK was not competitive for any of the tested problem sizes. LAPACK exhibited the best parallel scalability and achieved a good speedup

from dual-threading even for small problem sizes. The DCRM package only benefited appreciably from multi-threading for matrix sizes above 1000 x 1000. The PARDISO package did not scale well in the experiments.

For the 0^{th} and 6^{th} refinements of the spatial discretisation the total single-threaded running time was, respectively, 20 % longer and 43 % shorter when using the PARDISO package instead of the DCRM package. For the dual-threaded runs the total running times for the 0^{th} and 6^{th} refinements were, respectively, 21 % longer and 20 % shorter when using the PARDISO package instead of the DCRM package.

3.6.4. Conclusions

Speedup factors of 31-62 relative to the time that would be needed by an non-optimising implementation, were measured for the evaluation of Jacobians by the OESI. The speedups and the sparsities of the Jacobian matrix were limited for fine discretisations by a dense cluster of non-zero elements related to the moving mesh used in the appendix gap.

The multi-threading enabled by the OESI was found to speed up the solution of IVPs appreciably. But the maximum possible speedup did appear to have a relative low upper limit. Speedups between 1.45 and 1.67 were achieved for moderate problem sizes, i.e. for coarse discretisations, on a PC with a dual-core processor. But even with a supercomputer with very fast inter-processor communication the maximum achievable speedup for the relevant problem sizes appeared to be less than a factor of 5.

It was found that there were significant differences between the times needed by different linear algebra packages for performing LU-decomposition and that the differences depended on the problem size. The differences in the performance of the linear algebra packages also had significant impact on the total running time of simulations. No single linear algebra package appeared to be optimal for all problem sizes, and hence the OESIs capability to switch between different linear algebra packages is valuable.

3.7. Tests of the MUSSIM IVP solver

The *MUSSIM* IVP solver can use different Runge-Kutta schemes and has been designed to recycle as many Jacobians and iteration matrices as possible. Test have been performed to:

- Illustrate the accuracy and performance of different Runge-Kutta schemes for the Stirling machine model.
- Demonstrate that for one of the best performing Runge-Kutta schemes the accuracy, measured as the numbers significant correct digits in the power output and regenerator loss, is largely independent of the spatial discretisation.
- Show that the step size control and the recycling algorithms are effective at limiting the needed number of Jacobians and LU-decompositions.

To address these points the following tests have been performed:

1. Simulations of one unsteady cycle for the SM5 engine for the second refinement of the spatial discretisation in Table 1 were performed with seven different

Runge-Kutta schemes. For each scheme the tolerance for the error pr. step was varied from 10^{-1} to 10^{-11}. From the simulation results the accuracies and the time consumptions of the methods were extracted and visualised.
2. Simulations of both unsteady and periodic steady state cycles for the SM5 engine were performed for the different spatial discretisations in Table 1 using one of the Runge-Kutta schemes. For all the different spatial discretisations the tolerance for the error pr. step was varied between 10^{-1} and 10^{-11}. From the simulation results the accuracy as a function of the tolerance was plotted.
3. The distributions of step sizes, state changes, Jacobian updates and LU-decompositions during a cycle have been visualised for a simulation with the 2nd refinement of the spatial discretisation from Table 1.

3.7.1. Accuracy and performance of different Runge-Kutta schemes

The accuracy and performance of two explicit and five semi-implicit Runge-Kutta schemes were tested for the Stirling machine model. The tests were performed for the second refinement of the spatial discretisation in Table 1. The Runge-Kutta schemes were used in turn for integrating one periodic steady state cycle of the SM5 engine with relative tolerances for the IVP solver between 10^{-1} and 10^{-11}. For the semi-implicit methods the relative tolerance of the algebraic equation solver, which was used to find the stage solutions during each step, was kept two orders of magnitude smaller than the tolerance used for the IVP solver. The tests were run sequentially on a PC with 2 GHz AMD Athlon64X2 processor.

The electric power output, which is one of the primary performance indicators for the engine, was used to gauge the accuracy of the integration results. The power output predicted by the model is computed from integrations that are performed over the entire simulated cycle. At the same time the tolerance of the IVP method is a measure for maximum acceptable relative error in the variables of the model pr. integration step. So this test was really on how the final accuracy of integrated values depended on the tolerance for errors pr. step during the integration.

In the following the number of significant correct digits (SCD) in the power output calculated by the IVP solver is discussed. The number of significant correct digits is calculated as $-Log_{10}$ of the relative deviation from the exact solution. This accuracy is related solely to the accuracy of the IVP solver in the solution corresponding to the applied spatial discretisation. We do not yet consider the combined accuracy of the IVP method and the spatial discretisation in relation to the true solution to the model.

It was found that the relative differences were less than $2 \cdot 10^{-11}$ between the power outputs computed using a tolerance of 10^{-11} with three of the Runge-Kutta schemes. The three methods were:

1. The GERK scheme by Thomsen (2002). This is a semi-implicit Runge-Kutta method of order 3 with an embedded 4th order method. The first stage is explicit, since it is merely the values and derivatives at the start of the integration step.
2. An ESDIRK34 scheme by Alexander (2003). This scheme is also a semi-implicit Runge-Kutta method of order 3 with an embedded 4th order method and with an explicit first stage. The scheme has exactly the same structure as the GERK method by Thomsen but the coefficients are slightly different.

3. A 5th order explicit Runge-Kutta scheme by Cash and Karp with an embedded 4th order method found in Numerical Recipes (1997).

Extrapolations to an infinitely fine tolerance were made for these three methods, and the average of the extrapolated values was taken to be the true solution. This value was used for calculating the accuracies of all the solutions and the results are shown in Fig. 18.

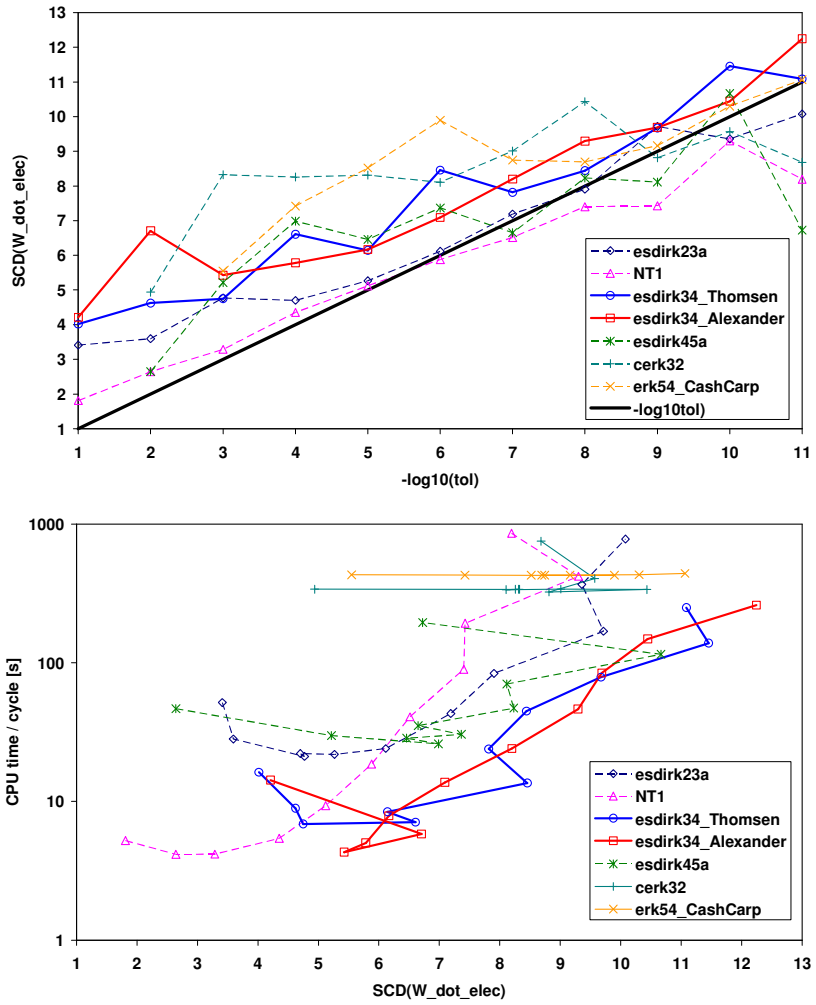


Fig. 18. Comparison of explicit and semi-implicit Runge-Kutta methods for simulation of one periodic steady state cycle for the second refinement of the spatial discretisation in Table 1. Top: Significant correct digits in the computed power output from a periodic cycle vs. -log10 of the relative tolerance pr. step. Bottom: CPU time needed for simulation of one periodic cycle vs. the number of significant correct digits in the computed power output (lower is better).

The top chart in Fig. 18 shows that for most of the schemes the numbers of significant correct digits in the computed power outputs were at least as good as *-log10* of the relative tolerance used in the IVP method. For some of the methods, however, the accuracy appeared to break down for very fine tolerances.

The bottom chart in Fig. 18 shows that the two 3rd order semi-implicit schemes by Thomsen and Alexander were consistently very competitive over a wide range of tolerances when one looks at the needed CPU time versus the achieved accuracy. Note that the vertical axis of the bottom chart in Fig. 18 is logarithmic. The differences between the times needed by the different semi-implicit schemes to produce a given accuracy were as large as an order of magnitude.

The CPU times needed by the explicit schemes were almost independent of the specified tolerances. This is because the equation system of the model is stiff. The explicit methods therefore needed to take many steps in order to maintain stability. From the bottom chart in Fig. 18 it appears that the equation system of the model is so stiff that the requirements for accuracy would have to be extreme in order for the explicit schemes to become competitive with the semi-implicit schemes.

3.7.2. Dependence of the integration accuracy on the spatial discretisation

In this test it was studied how the integration accuracy depended on the fineness of the spatial discretisation.

The Runge-Kutta scheme by Alexander (2003) was used for integrating both unsteady and periodic steady state cycles for the different spatial discretisations in Table 1. For each case integrations were performed with tolerances from 10^{-1} to 10^{-11}. For each case the power output and the regenerator loss were then extrapolated to the case of an infinitely fine tolerance for the IVP method. The extrapolated values were assumed to be the true solutions and were used for computing the number of significant correct digits in all the solutions. The regenerator loss was defined as the cycle averaged energy flux carried by the gas across the axial midpoint of the regenerator in the direction of decreasing temperature. The computed accuracies are plotted in Fig. 19.

Again the considered numbers of significant correct digits are related solely to the accuracy of the IVP solver for the solutions corresponding to each of the test cases. We are still not considering the combined accuracy of the IVP method and the spatial discretisation in relation to the true solution to the model.

Fig. 19 shows that the dependencies on the specified tolerances of the numbers of significant correct digits in the integrated values appeared to be largely independent of the spatial discretisation. There was possibly a weak tendency towards the integrated values being slightly less accurate for periodic steady state solutions than for unsteady solutions.

The numbers of significant correct digits in the computed regenerator losses appeared to be on average 1 to 2 digits less than *-log10* to the tolerance of the IVP method. To this it can be added that the amplitude of the instantaneous energy flux through the regenerator was of the order of magnitude 500 kW for these solutions. The regenerator loss is actually the integral mean value of this large oscillating energy flux. If the inaccuracies in the regenerator loss were scaled with the amplitude of the instantaneous energy flux through the regenerator, then the inaccuracies in the regenerator loss would be of approximately the same order of magnitude as the inaccuracies in the power output.

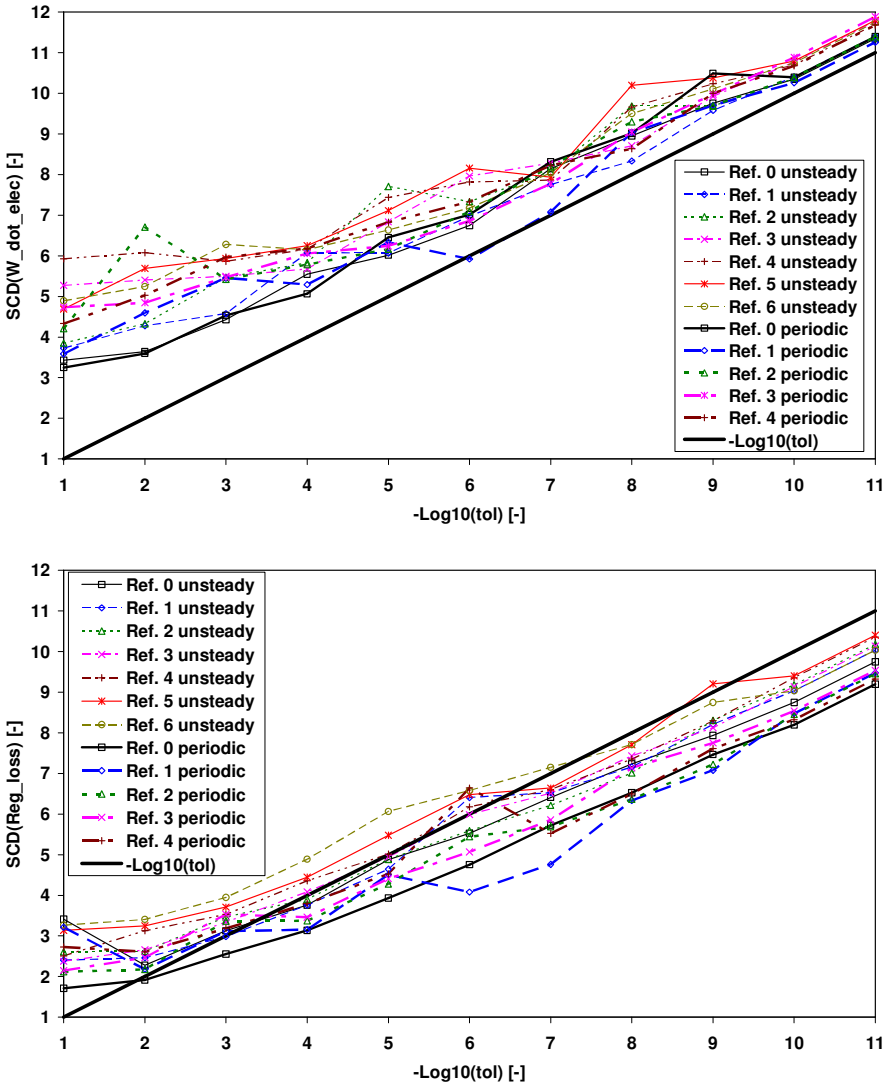


Fig. 19. The number of significant correct digits (SCD) for the power output (top) and the regenerator loss (bottom) versus the relative tolerance pr. step for one cycle of the SM5 Stirling engine model. Data is plotted for the different refinements of the spatial discretisation from Table 1 for both unsteady and periodic steady state solutions. The model was integrated using the ESDIRK34 method by Alexander (2003).

3.7.3. Step size control and recycling of Jacobians and iteration matrices

The distributions of step sizes, state changes, Jacobian updates and LU-decompositions during a cycle have been visualised in order to illustrate the effectiveness of the IVP solver and its step size control algorithm with respect to recycling Jacobians and iteration matrices.

The visualisation, which is shown in Fig. 20, was made for a simulation with the second refinement of the spatial discretisation from Table 1 and a relative tolerance for the IVP solver of 10^{-6}. The semi-implicit Runge-Kutta scheme by Alexander (2003) was used in the IVP solver.

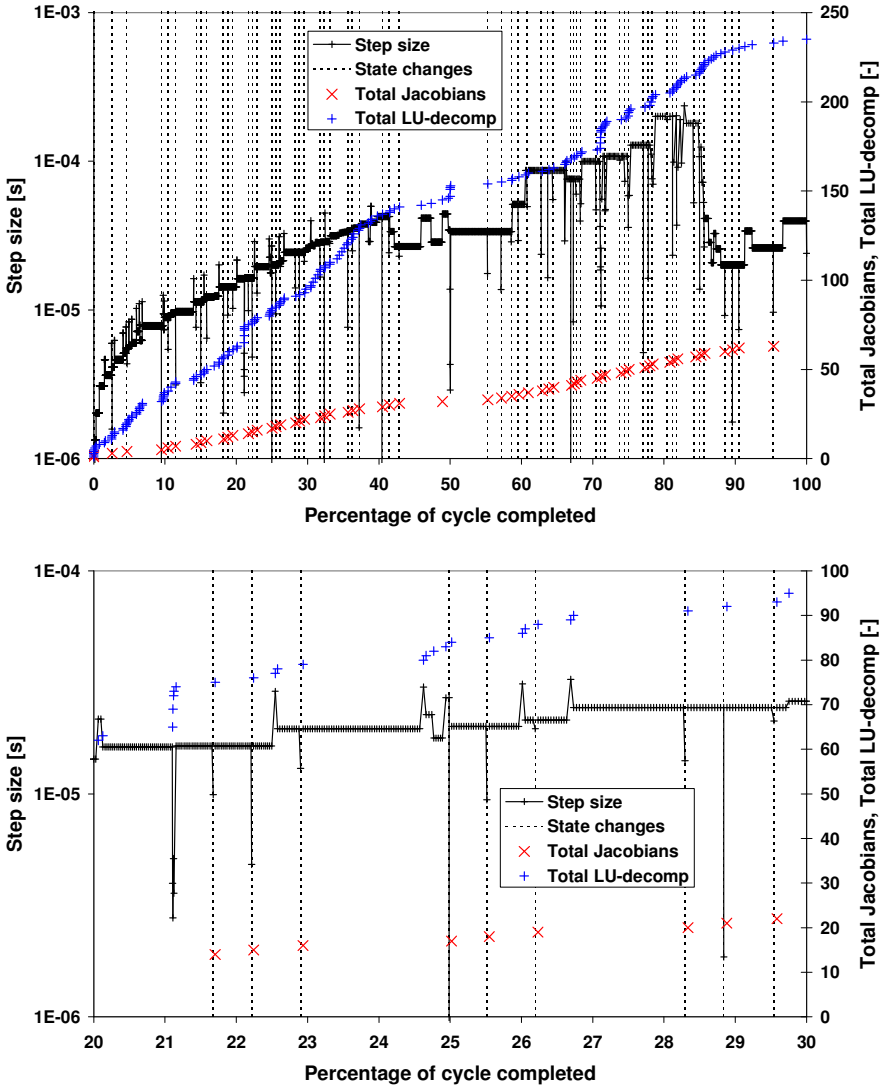


Fig. 20. Step sizes, state changes, Jacobian updates, and LU-decompositions during the integration of one cycle of the SM5 engine with the 2nd refinement of the spatial discretisation from Table 1 and a relative tolerance of 10^{-6} for the IVP solver. The top chart shows the entire cycle and the lower chart is a close up of the results where the cycle is between 20 and 30 % completed.

The total number of accepted steps during the integration was 3153. The solver discarded 51 steps due to too large error estimates. The step size was changed a total of 183 times out of which 51 times were due to discarded steps and 11 times were at state changes.

A total of 9612 solutions were computed to the implicit equation system of the semi-implicit Runge-Kutta scheme. On average 1.65 iterations were used pr. solution.

The solver expended 63 Jacobian updates during the cycle. Fig. 20 shows that one update was at the starting point of the integration, one was during flow reversal, and that the 61 remaining Jacobian updates were at state changes.

During the cycle 235 LU-decompositions of the iteration matrix were needed; Most of the decompositions occurred at step size changes as only 63 of the decompositions were due to Jacobian updates.

The above statistics and Fig. 20 show that the step size control algorithm generally enabled the solver to take many consecutive steps with the same step size. This enabled

the IVP solver to recycle each iteration matrix for an average of 67 iterations in an average of 41 solutions.

Fig. 20 also shows that the simple asymptotic error estimation in the step size control algorithm appeared to lead to sub-optimally large step size changes, so that it was sometimes only possible to take one step with a new step size before the step size was reduced again. So it appears that a minor speedup might be possible by using a better step size control algorithm.

For reference it can be noted that the two pulse tube cooler models, which are validated in the sections 3.9.3 and 3.9.4 below and which do not contain events, typically required approximately 5 Jacobians and 50 LU-decompositions when integration a cycle with a tolerance for the IVP solver of 10^{-7}.

In summary it appears that the step size control algorithm and the recycling algorithms of the IVP solver are effective at optimising the integration speed by minimising the numbers of Jacobians and LU-decompositions needed during integrations with semi-implicit Runge-Kutta methods.

3.7.4. Conclusions

From the numerical test on the accuracies and performance of different Runge-Kutta schemes for the IVP solver it was found that:

- The tested Runge-Kutta schemes produced results for integrated values that agreed very well; three of the schemes produced results that agreed up to 10.7 significant digits. The fact that independent methods produced results that agreed to such a high level of precision indicated that the implementation of the schemes is correct and that the schemes converge towards the same true solution.
- Two semi implicit schemes of order three by Thomsen (2002) and Alexander (2003) had the best performance of the tested schemes in terms of the CPU time needed to produce a given number of significant correct digits.
- The numbers of significant correct digits in the computed power outputs corresponded approximately to *-log10* of the relative tolerance for the error pr. step used in the IVP solver. That is, a tolerance of 10^{-7} for the IVP solver produced approximately 7 significant correct digits in the power output. The accuracies for the computed regenerator losses were generally 1-2 significant correct digits lower than *-log10* of the tolerance.
- The correlation between the numbers of significant correct digits in integrated values on the tolerance for errors pr. step appeared to be independent of the spatial discretisation.

It was also found that the algorithms for step size control and recycling in the IVP solver worked well when the solver was applied to the Stirling machine model. This means that:

- The equation system of the Stirling machine model is linear enough and the solutions are smooth enough to allow effective recycling of Jacobians.
- The IVP solver and its step size control algorithm do well at recycling Jacobians and iteration matrices.

3.8. Tests of the spatial discretisation of the modelling approach

A discretisation scheme is said to be convergent if it converges onto the correct solution when the discretisation is refined, i.e. when the number of subdivisions of the computational domain is increased. In solutions where discontinuities may be present three conditions are needed to ensure convergence: Stability, consistency, and conservativeness (Bauwens, 1993b):

- A discretisation scheme is said to be stable, if the numerical solution approaches a final limiting value when the discretisation is refined.
- A discretisation scheme is said to be consistent if its local behaviour approaches that of the underlying discretised differential equations when the discretisation is refined. For a stable discretisation scheme consistency should result from programming a correct version of the governing equations into the discretisation scheme.
- A discretisation scheme for a flow problem is said to be conservative, if it conserves mass and energy, even across jumps in the solution that do not vanish when the mesh size becomes small.

The stability, consistency, and conservativeness have been tested for the complete Stirling machine model described in section 3.4. In these tests of the spatial discretisation the results from the tests on the accuracy of the IVP solver were applied: The relative tolerances for the IVP solver were chosen, so that any errors due to the finite accuracy of the IVP solver were orders of magnitude smaller than the deviations due to the coarseness of the spatial discretisations.

The following tests were performed to document the convergence of the spatial discretisation:

1. The stability of the spatial discretisation was tested by numerical experiments for both unsteady and periodic steady state solutions by continuously refining the discretisation as shown in Table 1.
2. The consistency of the discretisation was tested by applying the model to two different flow problems with known analytical solutions:
   - Propagation of a temperature discontinuity.
   - The shock tube problem by Sod (1978).

   The test problems were studied directly in the heat rejector of the Stirling machine model from section 3.4.
3. The conservation of mass and energy, which by design should be enforced both locally and globally in the control volume based modelling approach, has been tested for both unsteady and periodic steady state solutions for the different discretisations in Table 1.

3.8.1. The stability of the spatial discretisation for unsteady solutions

In this section the results from a grid convergence study on the spatial discretisation for unsteady solutions are presented.

Unsteady solutions were calculated with the different discretisations from Table 1. The simulations were started from the initial distributions of temperature-, pressure-, and velocity described in section 3.4.3, i.e. from distributions where the initial values were functions of the axial space coordinate only. The stability of the discretisation

scheme was studied in terms of the convergence of the power output and regenerator loss onto finite values when the spatial discretisation was refined.

The tolerance for the IVP solver was kept fixed at 10^{-7}. According to the results of section 3.7 and the findings below, the errors due to the finite accuracy of the IVP solver should hence be insignificant compared to the deviations caused by the coarseness of the spatial discretisation. The results shown below should thus correctly reflect the properties of the spatial discretisation.

Fig. 21 shows the computed electrical power output and the regenerator loss for one cycle of the SM5 engine plotted against the inverse of the number of solid line control volumes in the spatial discretisation. Results are shown for the upstream approximation, the CILE method, and the van Leer method for interpolating enthalpies at control volume boundaries.

Please note that the high power outputs and the large negative regenerator losses do not correspond to a periodic steady state solution. They are artefacts caused by using the standard initial temperature-, pressure-, and velocity distributions from section 3.4.3.

Fig. 21 shows, that both the power output and the regenerator loss exhibited stability, i.e. they converged onto finite values when the discretisation was refined. The results for the three tested methods appeared to converge onto the same finite values. These finite values, which correspond to the case of infinitely many control volumes, are from here on denoted as the converged solutions.

The figure also shows that for a given fineness of the discretisation, the solutions calculated using the CILE and van Leer methods were significantly closer to the converged solution than the solutions calculated using the upstream approximation.

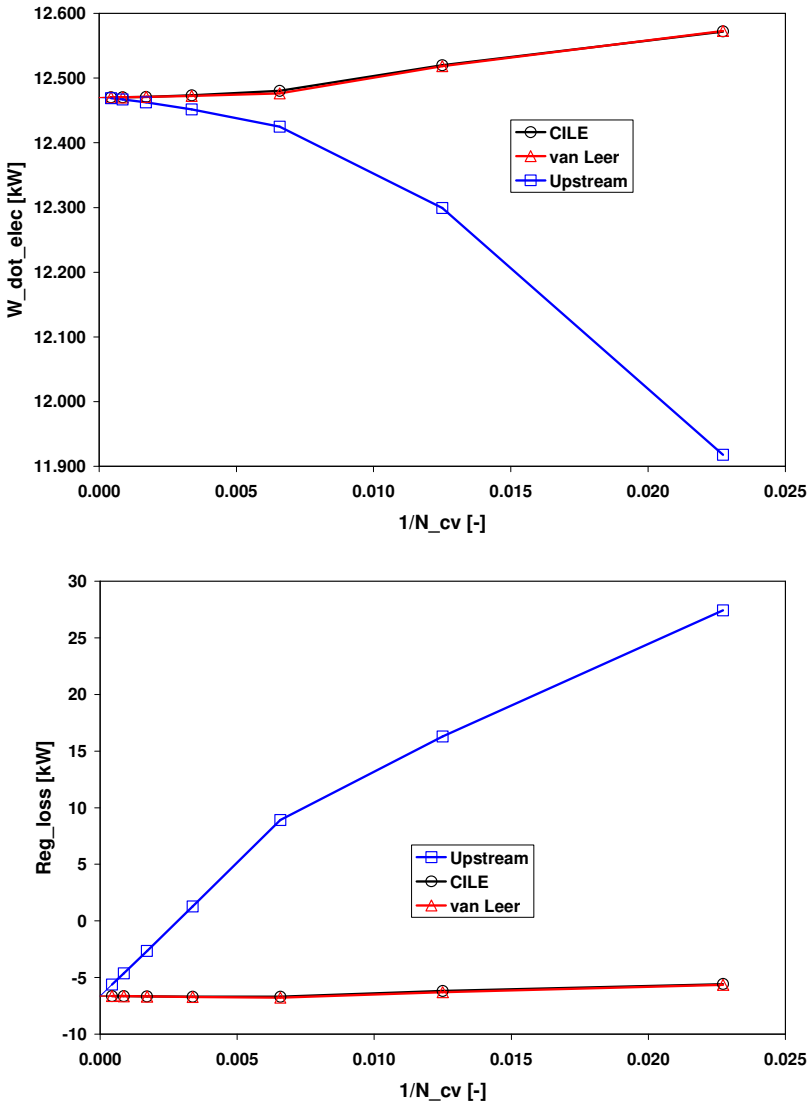


Fig. 21. Electrical power output and regenerator loss from one unsteady cycle of the SM5 Stirling engine plotted against the reciprocal of the number of solid line control volumes in the model. Results are shown for the upstream approximation, the CILE method, and the van Leer method for approximating enthalpies at control volume boundaries. Extrapolations to the case of infinitely many control volumes are shown as thin lines extending from the leftmost data points to the vertical axis.

The results for the CILE method were fitted with functions that have a constant term and a power function term. The constant term is intended to fit the value corresponding to the converged solution. The power function term is intended to fit the deviation from the converged solution. In the power function term it is assumed that the deviation is proportional to the reciprocal of the number control volumes raised to a constant power, i.e. that the deviations are a power function of the size of the control volumes.

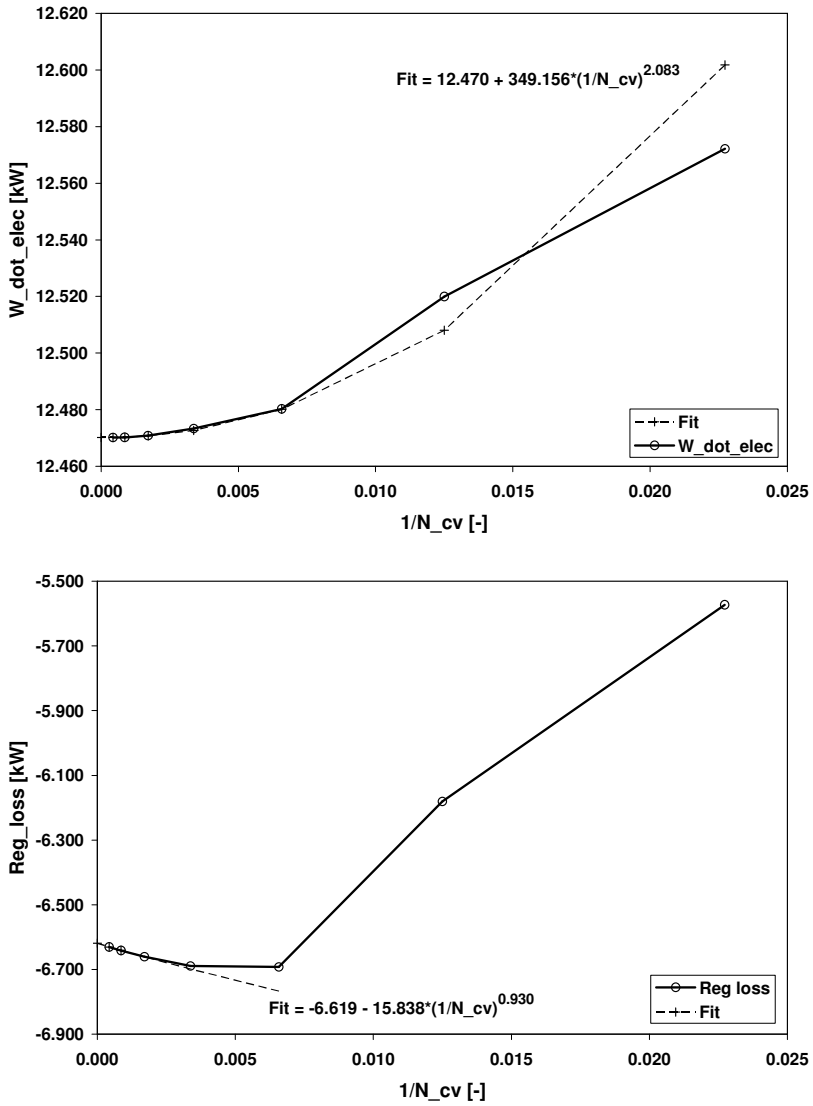


Fig. 22. Data and fitted functions for the electrical power output and the regenerator loss from one unsteady cycle for the SM5 Stirling engine plotted against the reciprocal of the number of solid line control volumes in the model. The CILE method was used for interpolating enthalpies at control volume boundaries.

Fig. 22 shows that the data was not fitted very well for the coarsest discretisations. But for the finer discretisations the power function model for the deviations from the converged solution appeared to be plausible. Parts of the deviations from the converged solution were caused by the coarsest discretisations not being able to properly resolve localised phenomena, such as localised matrix temperature oscillations in the ends of the regenerator which affect the regenerator loss (see the accompanying papers *A* and *B)*.

Fig. 23 shows the relative deviations from the converged solution for the upstream approximation, the CILE method, and the van Leer method plotted against the reciprocal of the number of solid line control volumes in the model.

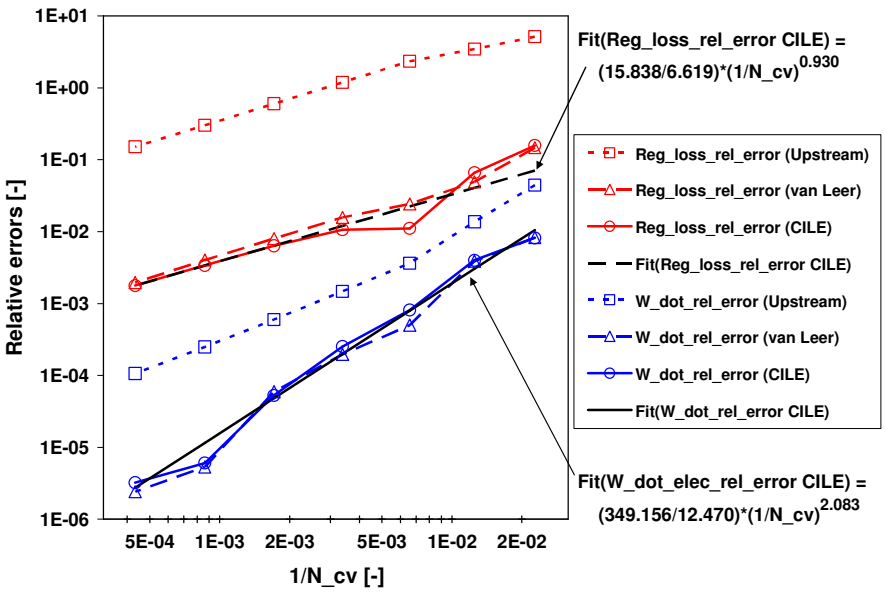


Fig. 23. The relative deviations in the electrical power output and the regenerator loss from one unsteady cycle for the SM5 Stirling engine due to the spatial discretisation. The deviations are plotted against the reciprocal of the number of solid line control volumes in the model.

Fig. 23 shows that the CILE and van Leer methods produced nearly identical deviations. The deviations for the upstream approximation were 1-2 orders of magnitude larger than the deviations for the CILE and van Leer methods.

The deviations for all three interpolation methods appear as nearly straight lines in the double logarithmic plot in Fig. 23, and hence the deviations can be approximated well by the power functions. For the CILE and van Leer methods the deviations in the power output were approximately proportional to the size of the control volumes raised to the second power. For the regenerator loss the deviations for the CILE and van Leer methods appeared to be approximately proportional to the size of the control volumes (raised to the first power).

It is important to note that for the CILE and van Leer methods the observed deviations all corresponded to modest fractions of the values in the converged solution. The largest observed deviation for the power output corresponded to approximately 0.8 % of the converged value. The largest deviation for the regenerator loss corresponded to approximately 16 % of the converged value.

For the CILE and van Leer methods it can hence be observed that even though the dominant terms in the deviations due to the spatial discretisation did not vanish very quickly when the discretisation was refined, the accuracy was still quite acceptable because the dominant terms in the deviations were not very large to begin with. In order to ensure approximately 3 significant correct digits for the power output it appeared sufficient to use the second refinement of the spatial refinement, i.e. 24 solid line control volumes pr. component.

Fig. 24 shows the numbers of significant correct digits in the power output and the regenerator loss for the upstream approximation, the CILE method, and the van Leer method as a function of the expended CPU time. The relative tolerance for the IVP solver was kept fixed at 10^{-7}.

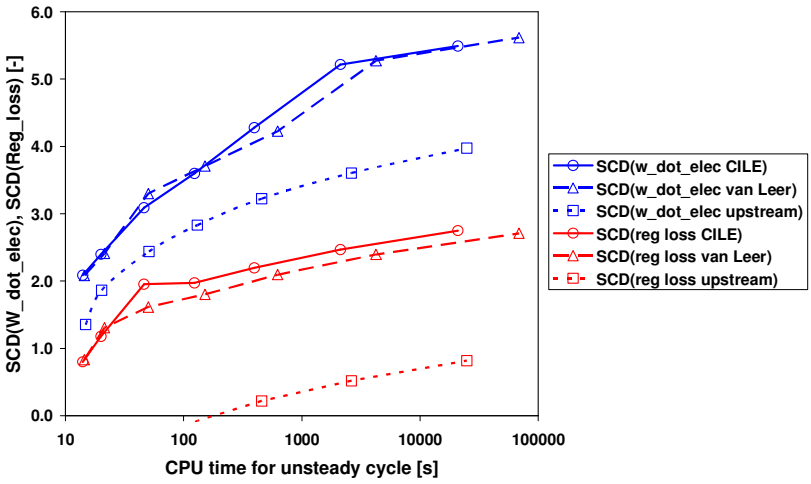


Fig. 24. Significant correct digits in the computed power output and regenerator loss from an unsteady cycle as a function of the expended CPU-time. Results are shown for the upstream approximation, the CILE method, and the van Leer method. The relative tolerance for the IVP solver was fixed at 10^{-7}.

From Fig. 24 it is clear that the upstream approximation produced the least accuracy pr. unit expended CPU time. The differences between the accuracies of the CILE method and the van Leer method were smaller but there was a tendency that the CILE method delivered better accuracy pr. unit of expended CPU time.

3.8.2. The stability of the spatial discretisation for periodic steady state solutions

In this section the results from a grid convergence study on the spatial discretisation for periodic steady state solutions are presented. In contrast to the results for unsteady solutions presented in section 3.8.1, the initial temperature-, pressure-, and velocity distributions depended on the spatial discretisation. All the results presented below were calculated using the CILE method for interpolating enthalpies at control volume boundaries

Fig. 25 illustrates the convergence of the spatial discretisation onto a periodic steady state solution. The figure contains both the computed values and fitted power functions of the type described in section 3.8.1. Results are included for the base discretisation and the first four refinements from Table 1. Results for the two finest refinements of Table 1 are not included because it would be very time consuming to find the corresponding periodic steady state solutions using a shooting method.

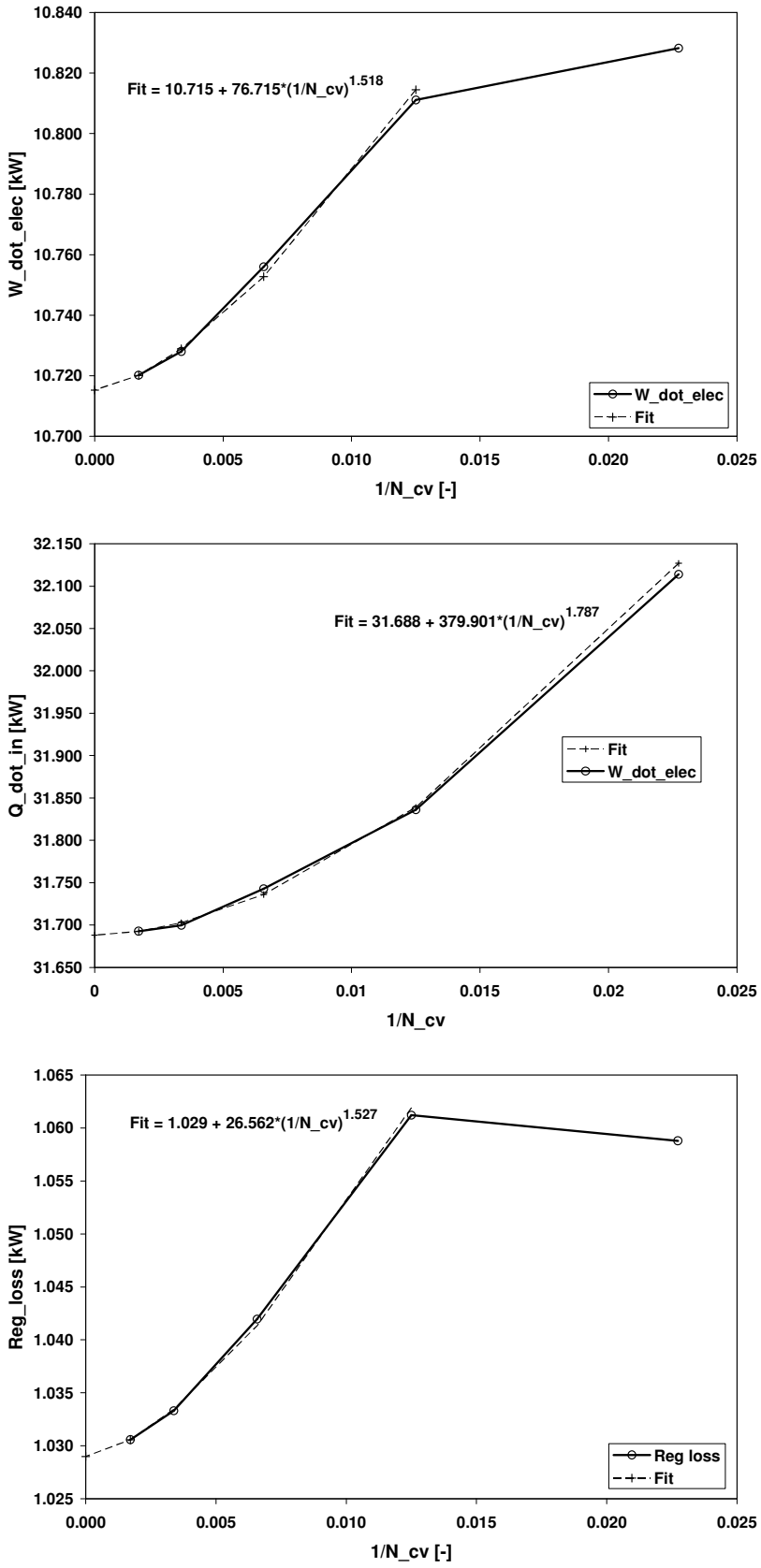


Fig. 25. Data and fitted functions for the electrical power output, heat intake, and the regenerator loss from periodic steady state solutions for the SM5 Stirling engine. The values are plotted against the reciprocal of the number of solid line control volumes in the model. The CILE method was used for interpolating enthalpies at control volume boundaries.

Fig. 25 shows that the discretisation of the modelling approach also appeared to be stable for periodic steady state solutions. As in Fig. 22 the correspondence between the data and the fitted functions was generally best for the finer discretisations.

Fig. 26 shows the relative deviations from the converged solution plotted against the reciprocal of the number of control volumes. The deviations were all approximated well by power functions whose values are proportional to the size of the control volumes raised to the power of approximately 1.5. For periodic steady state solutions it also appeared that it would be sufficient to use the second refinement with 24 solid line control volumes pr. component in order to ensure 2-3 significant correct digits for the heat intake and power output.

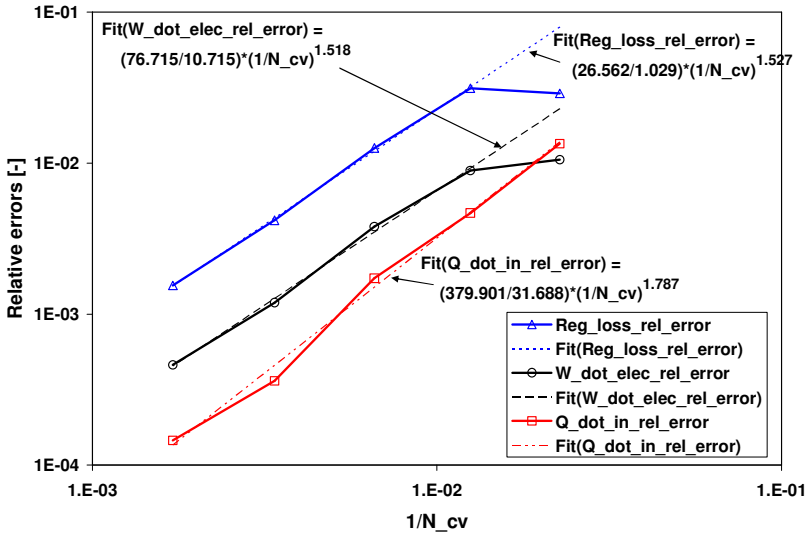


Fig. 26. The deviations in the computed electrical power output, heat intake, and regenerator loss from a periodic steady state solution for the SM5 Stirling engine due to the spatial discretisation. The deviations are plotted against the reciprocal of the number of solid line control volumes in the model.

3.8.3. The consistency of the spatial discretisation

In this section results are presented from tests which were performed to document that the discretisation scheme of the modelling approach is consistent, i.e. that the local behaviour of solutions approach the true behaviour of the underlying governing equations for compressible one-dimensional fluid flow when the discretisation is refined.

The consistency of the spatial discretisation was tested using test problems with propagating discontinuities. The discontinuities can be thought of as the limiting worst case scenarios for steep propagating wave fronts. If the modelling approach is consistent for problems with discontinuities then it will also be consistent for actual solutions to Stirling machine models. The consistency was tested for the following scenarios:

1. **Propagation of a temperature discontinuity:** This problem tests the consistency of the discretisation for steep propagating temperature wave fronts which may be generated at the inlets to discretised components during flow reversals.

2. **The Sod shock tube problem:** This problem tests the consistency of the discretisation with the governing equations for compressible flow when shocks and sonic velocities are present in solutions.

Both tests were carried out in the heat rejector of the Stirling machine model.

Propagation of a temperature discontinuity

The consistency of the modelling formulation was tested for the case of a temperature discontinuity propagating at constant velocity through a tube with uniform pressure. The following modifications were made to the Stirling machine model:

- The heat rejector was made 3 m long.
- The initial temperature was set equal to 273 K.
- The initial temperature in the leftmost manifold was locked and set equal to $0.9 \cdot 273$ K.
- The velocities in the heat rejector were set equal to 10 m/s and the velocities in the leftmost and rightmost boundary nodes were locked.
- The pressures in the manifolds at the ends of the heat rejector were locked.
- Friction and heat exchange with the wall were neglected.

The analytical solution to this test problem is a step in temperature which is smoothed only by molecular heat diffusion and which is travelling at 10 m/s from left to right through the tube. Fig. 27 shows the analytical solution and a number of numerical solutions calculated for different numbers of solid line control volumes in the tube. Snapshots were made at $t = 0.1$ s and $t = 0.2$ s. Results are shown for both the CILE method and the van Leer flux limiter for interpolating enthalpies at control volume boundaries.

Fig. 27 shows that the numerical solutions for both the CILE method and the van Leer flux limiter appeared to converge onto the analytical solution when the discretisation was refined. This means that the numerical solutions converged onto a solution where both the speed of propagation and the predicted rate of heat diffusion were correct.

The numerical solutions deviated from the analytical solutions in three different ways:

1. The step input was smeared as it entered the tube from the leftmost manifold volume.
2. Numerical diffusion lead to excessive smoothing of the front while it travelled through the domain.
3. The CILE method caused non-physical temperature oscillations.

The smearing due to the entry into the tube and due to numerical diffusion clearly diminished as the discretisation was refined.

With respect to numerical diffusion it can be seen that even for the coarsest discretisations there was only moderate difference in the gradients between the two snapshots for both the CILE and the van Leer method. This indicates that the influence of numerical diffusion was moderate for the smeared front.

The non-physical oscillations caused by the CILE method also diminished as the discretisation was refined. The overshoot that preceded the front was made narrower by the refinements of the discretisation. Once the discretisation was fine enough the resulting large temperature gradients at the narrow overshoot caused the overshoot to be smoothed away by heat diffusion.

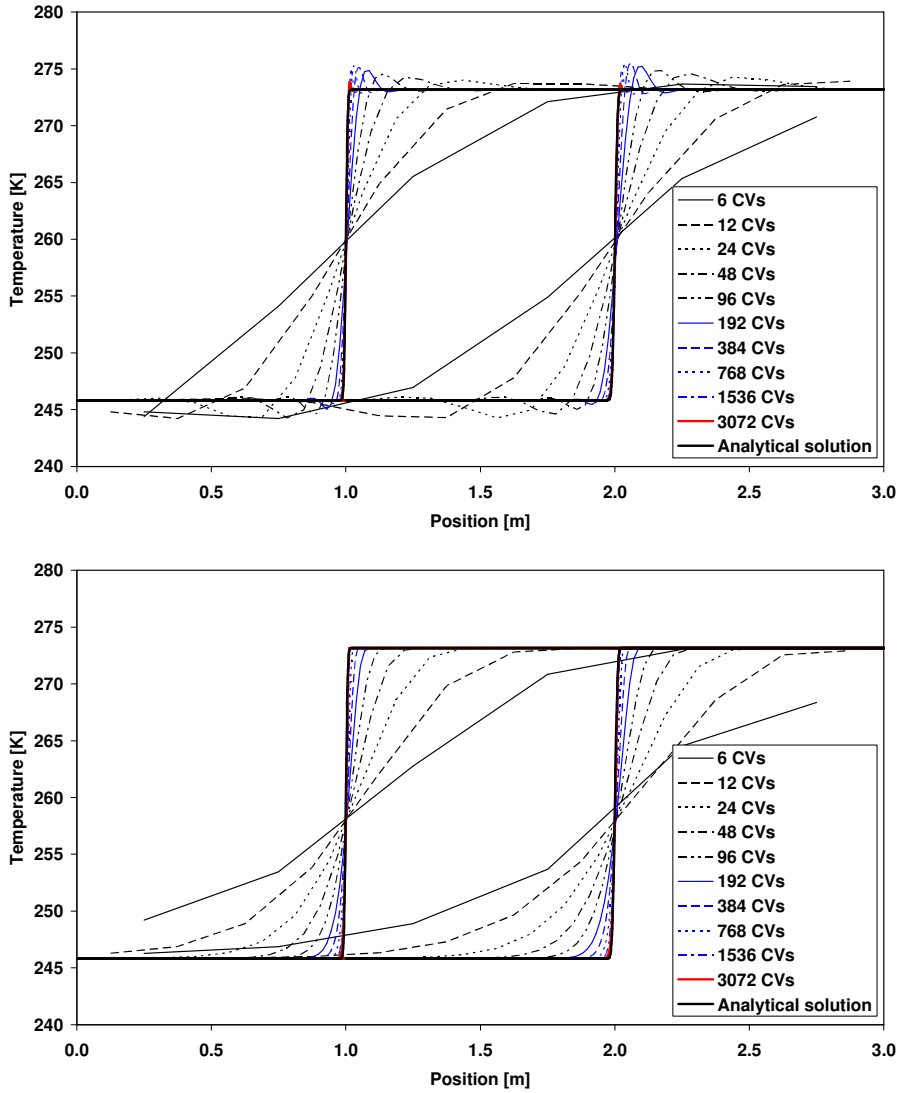


Fig. 27. Temperature discontinuity travelling through a tube discretised into different numbers of control volumes. Snapshots were made at t = 0.1 s and t = 0.2 s. Boundary enthalpies were interpolated using the CILE method (top chart) and the van Leer flux limiter (bottom chart).

The Sod shock tube problem

The shock tube problem by Sod (1978) is a widely used test problem for compressible flow codes. The problem setup consists of a channel which is divided into two halves by a membrane. The fluid on both sides of the membrane is initially at rest. On the left hand side of the membrane the pressure and density are, respectively, 10 times and 8 times higher than on the right hand side. The initial conditions used here in this test, which was conducted with ideal gas helium in the channel, are illustrated in Fig. 28.

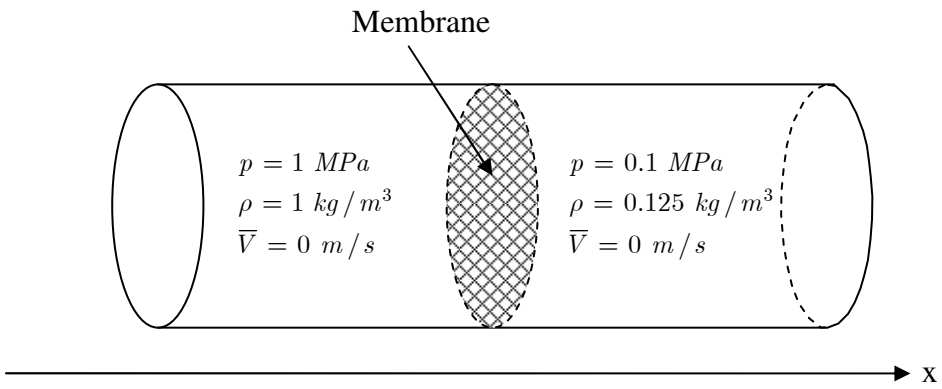


Fig. 28. Initial conditions for the Sod shock tube problem.

At time zero the membrane is burst and three different types of discontinuities form and begin to propagate through the tube. The following descriptions of the discontinuities are based on Fedkiw et al (1998):

1. A shock in the form of an increase in pressure and density begins to travel to the right at high velocity. The shock propagates by a mechanism by which strong impulses move faster than weak ones. The "top" of the shock wave is hence travelling faster than any smearing that precedes the front and hence the front of the shock wave is self sharpening. The speed of propagation is given by the difference in influx and outflux of conserved quantities to the region surrounding the shock. Therefore the shock will only propagate at the correct speed if the conservation laws are fulfilled across the shock
2. A rarefaction zone expands towards the left as the gas in the left end of the tube begins to expand. The front edge of the rarefaction propagates at the local speed of sound.
3. A contact discontinuity, which is a localised jump in density, travels towards the right at the bulk convection speed. The density jump is a remnant from the initial discontinuity at the membrane. There is no pressure difference or velocity difference across the contact and hence the only mechanism of transport across the contact is molecular diffusion. The density jump is hence similar to the propagating temperature discontinuity simulated above but the mechanism which forms it is different. Any spurious oscillations and inaccuracies due to the initial transient at the location of the membrane will persist at the contact.

The initial conditions of the shock tube problem were applied to the control volumes in the heat rejector. The following additional changes were made to the Stirling machine model:

- The heat rejector was modified to be a 6 m long tube.
- The properties in the outermost nodes of the heat rejector were locked.
- Friction and heat exchange with the wall were neglected.

An analytical solution to the test problem at 1 ms after the bursting of the membrane was calculated using a program made available by Chandrashekar (2006). The solution is for the Euler equations, which correspond to the Navier-Stokes equations with zero viscosity. Fig. 29 shows the pressure-, density, and velocity distributions of this analytical solution together with numerical solutions calculated with different numbers of solid line control volumes for the CILE method. Fig. 30 shows the corresponding results calculated using the van Leer method.

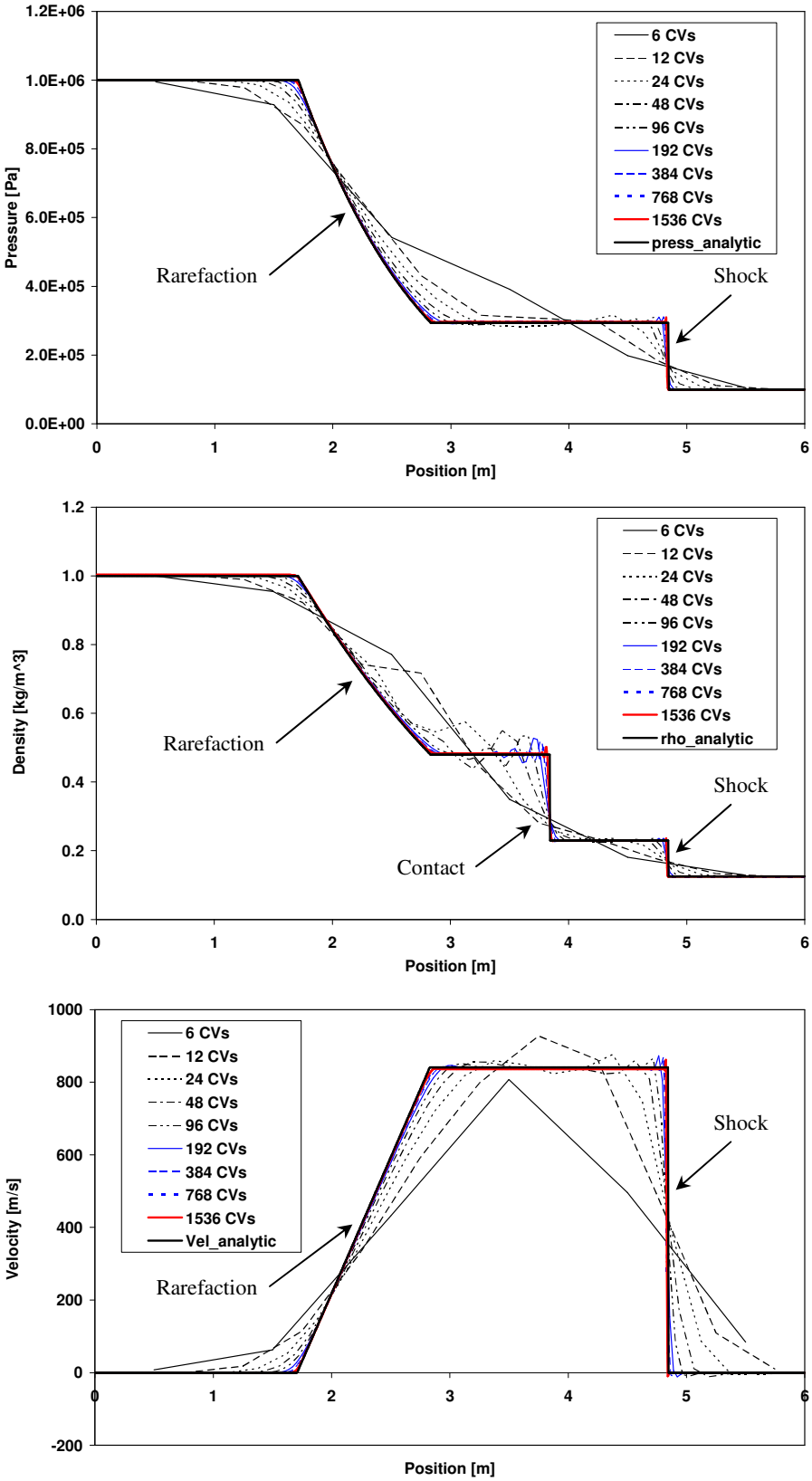


Fig. 29. Pressure, density, and velocity distributions for the Sod shock tube test problem. Results were calculated using the CILE method for interpolating pressures and enthalpies.

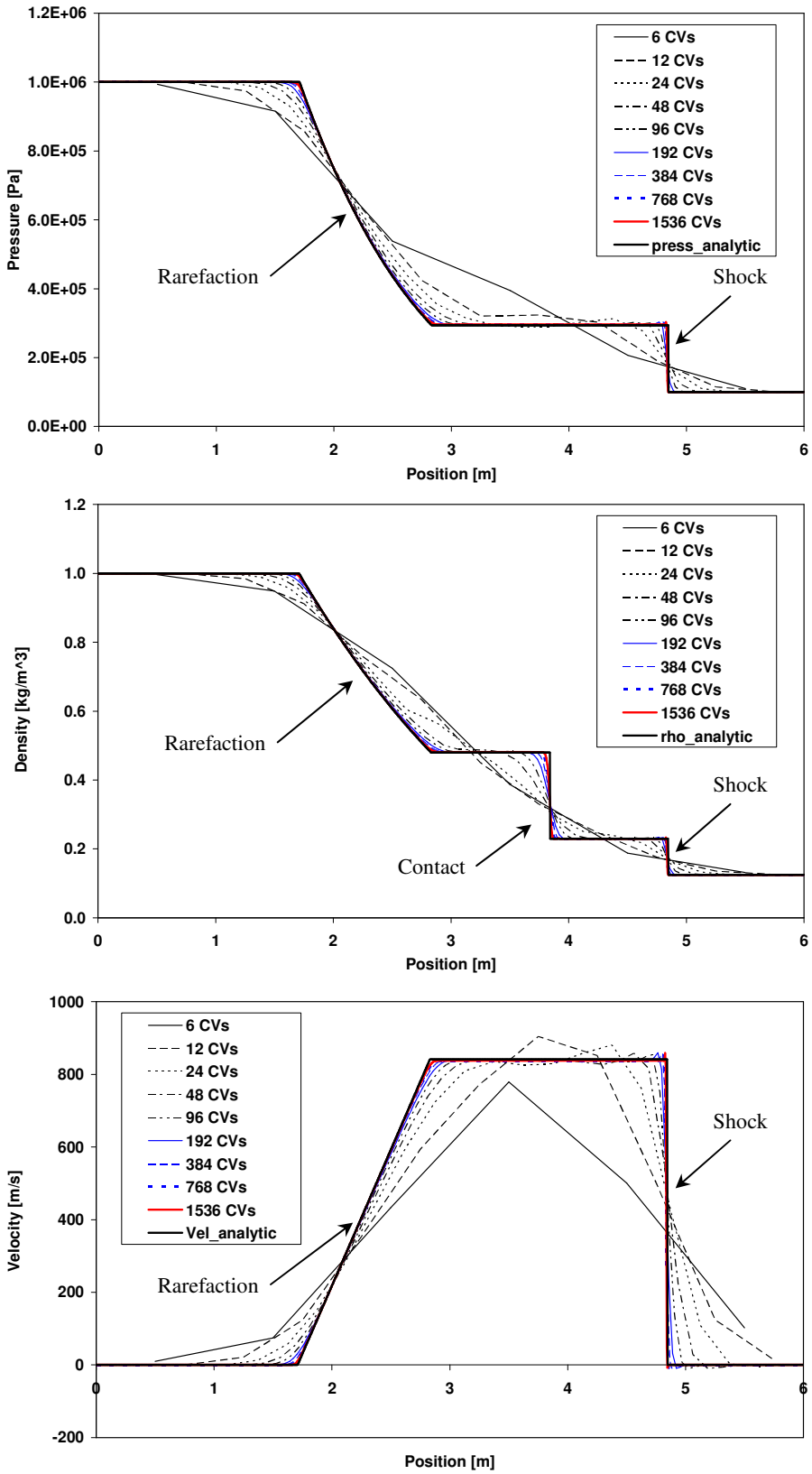


Fig. 30. Pressure, density, and velocity distributions for the Sod shock tube test problem. Results were calculated using the van Leer flux limiter for interpolating pressures and enthalpies.

Fig. 29 and Fig. 30 show that the distributions of pressure, density, and velocity in the numerical solutions appeared to converge onto the analytical solution when the

discretisation was refined. The shock was travelling at the correct speed and this means that the conservation equations were fulfilled across the shock. The rarefaction was also propagating at the correct speed of sound.

The numerical solutions in Fig. 29, which were calculated using the CILE method, contained significant oscillations. But the oscillations appeared to diminish when the discretisation was refined. There appeared to be no evidence for any of the spatial discretisations that the shocks should destroy solutions calculated using the CILE method. Fig. 30 shows that the oscillations and overshoots could be almost eliminated by using the van Leer flux limiter for interpolating pressures and enthalpies.

3.8.4. Conservation of mass and energy

The test results for the shock tube problem showed that shocks propagate at the correct speed and this means that the conservation laws are fulfilled locally across the shock. It has also been tested that mass and energy is conserved globally for the Stirling machine model.

It is always possible to test for conservation of mass by monitoring the sum of the masses in all the solid line control volumes.

Conservation of energy can be verified for periodic steady state solutions by looking at an energy balance for a control volume surrounding the entire simulated machine, i.e. by looking at the heat and work flows in and out of the simulated machine. For periodic steady state solutions the rate of energy accumulation inside the machine should be zero. Note that zero energy accumulation is not used as an equation in the conditions for periodic steady state given in section 3.2.8. But if a solution is a periodic steady state solution and if all aspects of the model formulation are energy conserving then zero energy accumulation will automatically follow.

Table 6 shows results for conservation of mass and energy from the solutions calculated with the CILE method for the tests of the stability of the spatial discretisation.

			Unsteady solutions	**Periodic steady state solutions**	
			(m_max-m_min)/m_avrg	(m_max-m_min)/m_avrg	dE/dt / Q_dot_in
Refinement	N_cv	N_ODEs	[-]	[-]	[-]
0	44	214	3.1E-09	1.1E-09	-3.4E-09
1	80	380	4.7E-09	9.1E-09	-3.0E-06
2	152	712	2.6E-09	3.4E-09	-1.5E-06
3	296	1376	1.9E-09	2.9E-09	-5.5E-07
4	584	2704	1.2E-09	3.0E-09	-1.3E-06
5	1160	5360	6.2E-10		
6	2312	10672	1.2E-08		

Table 6. Summary of results for the fluctuations of the total mass during both unsteady and periodic steady state solutions for different spatial discretisations and results for energy accumulation from periodic steady state solutions.

Table 6 shows that mass was conserved globally to a very strict tolerance for both the unsteady and periodic steady state solutions.

The rates of energy accumulation in the periodic steady state solutions were also insignificant compared to the rate of energy flow through the engine. This indicates that all aspects of the modelling formulation, including the energy transport at the moving mesh, are energy conserving. It also indicates that the periodic steady state solutions were very close to being true periodic steady state solutions.

3.8.5. Conclusions

Tests on the stability of the spatial discretisation scheme showed that the scheme was stable for both unsteady and periodic steady state solutions. The tests also showed that the CILE method, the upstream approximation, and the van Leer method for interpolating enthalpies and pressures at boundaries all converged onto the same solution when the discretisation was refined.

Using the CILE or van Leer methods instead of the upstream approximation improved the accuracy of the solutions. The CILE and van Leer methods also produced better accuracy pr. expended unit of CPU time than the upstream approximation. The CILE method delivered slightly more accuracy pr. unit of CPU time than the van Leer method.

When the CILE or van Leer methods were used a discretisation with 24 control volumes pr. component appeared sufficient to produce 2-3 significant correct digits in the computed heat intakes and power outputs, i.e. 2-3 significant correct digits of the values which would result from using infinitely many control volumes pr. component.

The tests of the consistency of the spatial discretisation showed that the discretisation appeared to be consistent even when shocks and sonic velocities were present in the solutions. Both numerical diffusion and non-physical oscillations in the solutions diminished when the discretisation was sufficiently refined.

It must be noted, however, that for the CILE method the non-physical oscillations at the discontinuities and shocks in the test problems persisted to discretisations which were significantly finer than the discretisations which it would typically be practical to use in a Stirling machine model. But periodic solutions to models created using the modelling approach presented here will not contain any propagating discontinuities due to the way in which components are connected. They will at most contain relatively steep faced temperature waves. The tests on the stability of the spatial discretisation also showed that using the CILE method for relatively coarse discretisations produced performance values which were accurate to several significant digits. Hence it was conclude that either the oscillations introduced into solutions to the Stirling machine model by the CILE method are small or at least they are not very important to the predicted machine performance.

But if a situation arises where there are doubts as to whether the non-physical oscillations introduce significant errors into solutions then it appears advisable to check if switching to the van Leer method changes the solutions significantly.

In this context, where we seek a sensible balance between accuracy and needed computational efforts for models which already contain the significant assumption of one-dimensional flow, the oscillations introduced by the CILE method appear to be an acceptable trade off for increasing the speed of simulations.

It was also shown that mass was conserved to very strict tolerance for both unsteady and periodic steady state solutions. Conservation of energy was also confirmed for periodic steady state solutions by looking at the rate of energy accumulation in the machine. This test also confirmed that the periodic steady state solutions used in the tests were accurate steady state solutions.

3.9. Validation of models against experimental data and simulation results

In section 3.9.1 simulation results, which were computed using the Stirling machine model described in section 3.4, are compared to experimental data for the SM5 engine.

In section 3.9.2 simulation results for the same Stirling machine model are compared to experimental data for a commercially available free piston Stirling cooler from Twinbird. The simulation results for the Twinbird cooler are also compared to simulation results calculated with the *PROSA* software.

In sections 3.9.3 and 3.9.4 simulations results for two pulse tube cooler models based on the modelling approach presented above are compared to simulation results for two pulse tube cooler models created using the *Sage* software.

3.9.1. Validation against experimental results for the SM5 Stirling engine

The Stirling machine model described in section 3.4 has been validated by comparing simulation results to experimental data for the SM5 Stirling engine at 28 distinct operating conditions.

The experimental data

The experimental data was provided for use in this thesis by Carlsen (2006). The experiments with the SM5 engine were performed with the engine mounted on a test bed where the following quantities were measured during steady state operation:

- The electrical power output.
- The mass flow rates and temperatures of the cooling water flows which cooled the engine and its generator.
- The temperatures on the heater tubes, the temperatures at different locations in the flue gas system, and the temperature of the ambient air surrounding the test bed.
- The minimum and maximum pressures during the cycle.
- The mass flow of fuel (natural gas) into the system.
- The chemical composition of the flue gasses.

In the experiments the geometric mean pressure in the engine was varied between 53 and 84 Bar, the mean cooling water temperature was varied between 34 °C and 66 °C, the average heater tube temperatures were varied between 558 °C and 736 °C, and both helium and nitrogen were tested as working gases. Using nitrogen instead of helium results in much larger flow friction and a larger heat capacity pr. unit volume for the gas in the engine. The experiments hence covered a wide range of operating conditions.

By looking at overall energy balances for the test bed at steady state operation it was determined that the measurements contained some inaccuracies. When the energy flows in and out of the setup were determined from the measured data then the overall energy balance was off by up to 3.5 kW for some of the measurements. The inaccuracies in the measurements were believed to be partly due to problems with measuring true average temperatures for the flue gas, due to problems with maintaining a constant cooling water temperature for long enough to reach true steady state operation, and due to inaccuracies in the estimations of the heat losses from the burner system to the air in the lab.

Given the amount of data, however, it was assessed that the data was sufficiently accurate to see if the model was able to capture the dependence of the performance of the engine on its operating conditions.

The measured data defined 28 different operating points by the heater tube temperatures and the cooling water temperatures. The rate of energy absorption by the heater tubes was derived from the rate of fuel consumption of the burner, the measured flue gas data, and approximate calculations of the heat losses from the burner system to the air surrounding the test bed. The electric power output at each operating point had been measured and the efficiency of the engines built in generator was known to be approximately 90 %. The shaft power produced at each operating point could hence be calculated by dividing the measured electrical power outputs with the generator efficiency.

Hence data was available for a comparison of 28 sets of measured and calculated values for the shaft power output and the thermal efficiency of the engine. The thermal efficiency was defined as the ratio between the shaft power output and the heat input to the heater tubes.

The model setup

The Stirling machine model computed the work that the gas exerted on the work piston and on the displacer. In order to be able to compare measured and calculated values for the shaft power the simulation results had to be corrected for:

- Friction due to the seals sliding against the cylinder wall.
- Friction in the crank mechanism.

Approximate values for the friction forces at the seals between the cylinder wall and the piston and displacer were known from previous experiments. These forces were used directly in the Stirling machine model for computing the piston rod forces from the forces exerted on the piston and displacer by the gas. The heat produced by the friction at the sliding seals was added directly to the seal and the cylinder wall in the simulations.

The friction in the crank mechanism itself was needed in the model for computing the shaft power from the power transmitted through the piston rods. The magnitude of the friction in the crank mechanism was not well known. But it was known that, as long as the operating speed of the engine and the temperature of the grease in the bearings were constant, then the friction torque in the bearings of the crank mechanism would be an approximately linear function of the forces on the bearings. It was hence assumed that the friction in the crank mechanism was a linear function of the average forces transmitted to the crank mechanism by the piston rods.

The shaft power outputs were known from the experiments and the average forces transmitted by the piston rods to the crank mechanism were known from simulations of the engine. The crank mechanism friction torque was then fitted as a linear function of the piston rod forces so that the resulting losses minimised the differences between the measured and calculated shaft power at the 28 operating points. The fitted linear function for the crank mechanism friction torque was then incorporated into the Stirling machine model.

The Redlich-Kwong equation of state was used in the model for describing the helium and nitrogen because of the high pressures in the SM5 engine. For the simulations

performed with nitrogen it did make a significant difference in the computed power output when the Redlich-Kwong equation of state was used instead of the ideal gas equation of state.

The spatial discretisation corresponding to the second refinement in Table 1 was used for the simulations. The computed power outputs and heat intakes should hence correspond to the converged solution to the model with an accuracy of 2-3 significant correct digits. Hence the deviations due to numerical inaccuracies in the solutions should be insignificant compared to the deviations due to the assumptions and simplifications in the one-dimensional model.

Comparison of experimental data and simulation results

It was chosen to plot the shaft power outputs and the shaft efficiencies against the Carnot efficiency which was calculated from the hottest heater tube temperatures and the average cooling water temperatures. This allowed the data from the experiments to be plotted against a single axis and it also spread out the data points nicely. The comparison of the experimental data to simulation results is shown in Fig. 31.

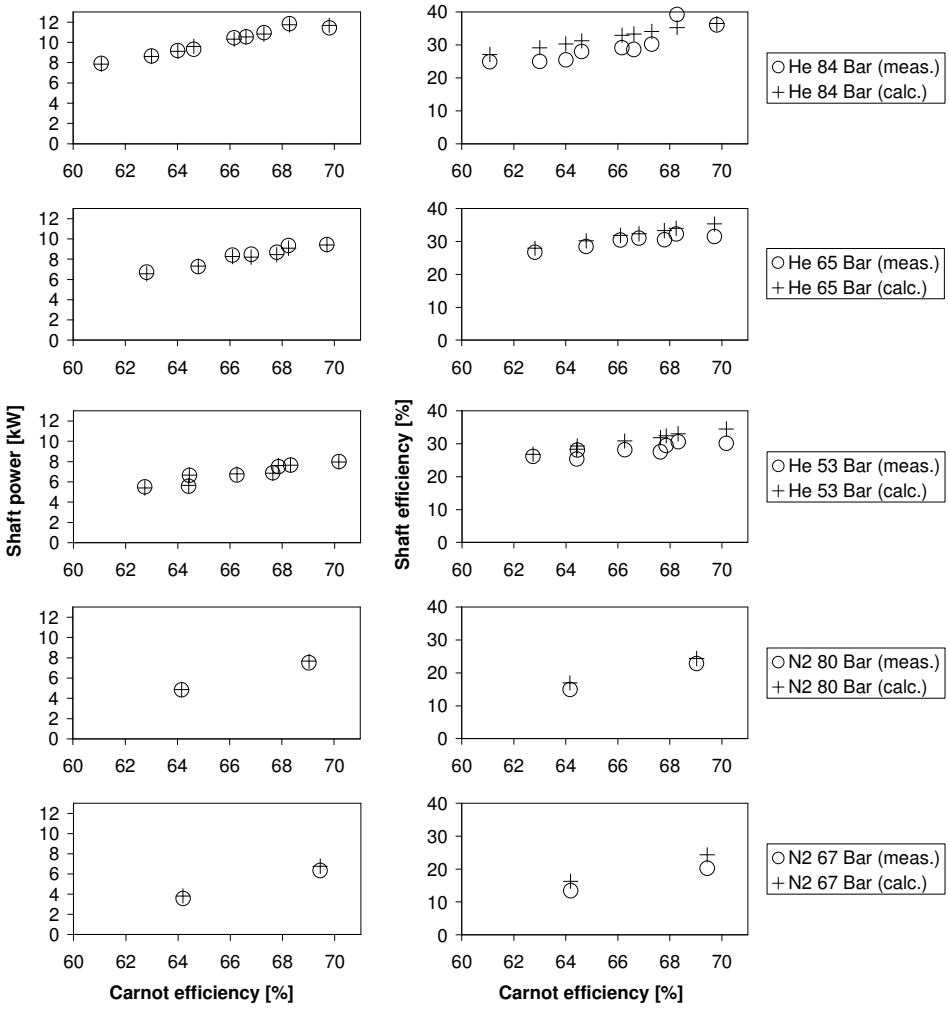


Fig. 31. Comparison of experimental data (O symbols) and simulation results (+ symbols) for the SM5 engine. The left column of charts show measured and calculated shaft powers. The charts in the right column show the shaft efficiencies. The rows of charts correspond to the pressures and working gasses shown in the legends on the right. The data is plotted against the Carnot efficiency calculated from the average cooling water temperature and the maximum heater tube temperature.

Fig. 31 shows that the model predicts the work output of the engine with good accuracy (deviations < 3 % for helium, deviations < 7 % for nitrogen) over the tested range of operating conditions. The efficiency of the engine predicted by the model is generally a little higher than the experimental results (deviations up to approx. 20 % for both helium and nitrogen).

The inaccuracies in the experimental data are apparent in Fig. 31. The measurement that shows an efficiency of 40 % in the topmost chart for the efficiency, for instance, is not believed to be accurate.

Sources of uncertainties

Uncertainties were introduced into the results at three levels:

- Uncertainties in the experimental results.
- Uncertainties in the thermodynamic analysis.
- Uncertainties in the estimation of the mechanical friction.

The uncertainties in the experimental results were significant as discussed above.

The uncertainties in the thermodynamic analysis of the engine were introduced by the simplifications inherent to a one-dimensional model and by uncertainties related to the empirical correlations used for computing heat transfer and flow friction. The correlations used in the simulations of the engine were developed for steady state flow. The simplifications made in the description of the multi-dimensional heat conduction in the steel of the engine also contributed with uncertainties.

Both the calculated power output and the calculated efficiency were sensitive to the choices of empirical correlations for calculating heat transfer and flow friction in the regenerator and for calculating heat transfer in the displacer clearance gap.

The correlation for flow friction in the regenerator is especially important when nitrogen is used as working gas. In the simulations that produced the results in Fig. 31 a correlation by Thomas (2000) for data by Kühl was used for computing friction in the regenerator. For these simulations using a different regenerator correlation could offset the calculated power outputs by up to a couple of kilowatts. However, the flow friction through this particular type of regenerator has been studied in separate experiments by Marcus-Moeller and Mortensen (2005), and hence it was known that the friction correlation by Kühl is reasonably accurate for the regenerator in the SM5 engine.

The correlation used for heat transfer in the regenerator directly affected the regenerator loss. The regenerator loss mainly affected the heat intake of the engine but it also had an influence on the power output. In Fig. 31 the choice of regenerator heat transfer correlation hence mainly affected the calculated efficiencies. The regenerator loss was significantly larger with nitrogen than with helium and hence the results for nitrogen as working gas were more sensitive to the heat transfer correlation than the results for helium. In the simulations that produced the results in Fig. 31 a correlation by Thomas (2000) for data by Tanaka was used for computing heat transfer in the regenerator.

The correlation for heat transfer inside the displacer clearance gap affected the amount of energy transported along the gap from the expansion volume and down to the cooler. This loss mainly affected the heat intake and, hence, the efficiency of the engine. In this study a heat transfer correlation by Huang and Berggren (1986) was used in the gap.

Due to the way that the mechanical friction was estimated and included in the model it should tend to reduce the errors in the predicted work output. But because the mechanical friction was estimated as a simple linear function of the bearing forces and because such a wide range of operating conditions was studied it appears unlikely that the mechanical friction should have been able to hide any significant model errors.

Conclusion
The agreement with experimental data over a wide range of operating conditions was good, even considering that the friction in the crank mechanism was adapted to the experimental data.

3.9.2. Validation against experimental results for the Twinbird Stirling cooler and PROSA simulation results

The Stirling machine model from section 3.4 was also validated by comparing simulation results from the model to experimental data for a commercially available free piston Stirling cooler (FPSC) by Twinbird (2006). A comparison was also made to results from a *PROSA* model of this cooler. For convenience the Stirling machine model from section 3.4 is denoted as the *MUSSIM* model in the rest of this section.

The Twinbird FPSCs are sold integrated into portable 25 litre cool boxes (SC-C925) for food and beverages. The vertical inner walls of the cool box are cooled by a thermosyphon connected to the FPSC. A photo of a Twinbird FPSC and a drawing illustrating its construction are shown in Fig. 32.

The Twinbird FPSC is a hermetically sealed β-type free piston Stirling cooler. The FPSC has no crank mechanism. Instead the work piston and the displacer are mounted on flat springs. The work piston is driven to oscillate in its suspension by a linear motor. The displacer is animated by pressure forces, which have two main contributions:

- The oscillating pressure pushes on different sizes of areas on the ends of the displacer due to the rod which connects the displacer to its flat spring.
- The flow friction for the oscillating gas flow through the regenerator and heat exchangers causes an oscillating pressure difference between the ends of the displacer.

Two Twinbird SC-C925 cool boxes were used for an experimental validation of the Stirling machine model.

The FPSC was removed from one of the boxes. The FPSC was disassembled so that its internal geometry could be studied in order to generate input data to the Stirling machine model. The moving masses, the rigidities of the flat springs, and the dimensions of the linear motor were also measured.

The other cool box was used for conducting experiments to determine sets of values for the temperatures at the heat absorber and heat rejector, the heat flow to the heat absorber, and the electric power input to the linear motor.

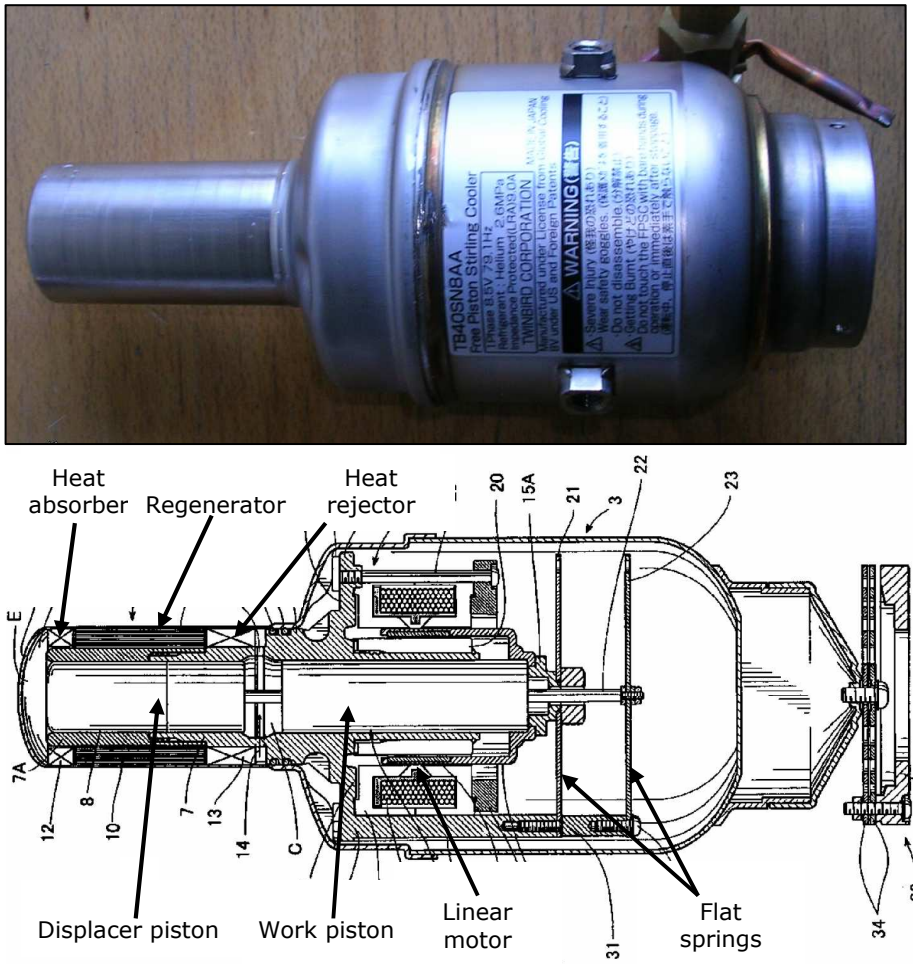


Fig. 32. Photo of the Twinbird FPSC (top) and sketch of the Twinbird FPSC (bottom). The sketch is from the patent US 6,857,267 B2.

Experimental data

The experiments were conducted with the FPSC still mounted in the cool box. The cool box was instrumented so that the following values could be determined:

- The electrical power input into the FPSC through the wires that connected the FPSC to the electronics of the cool box.
- The temperatures on the outer casing of the FPSC at the locations of the heat absorber and the heat rejector.
- The temperature on a vertical inner wall in the cool box.
- The temperature of the ambient air.

Experiments were performed in order to determine the overall conductivity of the cool box, i.e. to determine the net heat transport between the inner wall of the box and the ambient air as a function of the temperature difference between the inner wall and the ambient air.

In these experiments the power supply to the cool box was disconnected and a light bulb was inserted into the box. By supplying power to the light bulb the cool box was heated internally. The amount of power supplied to the light bulb and the temperature difference between the inner wall and the ambient air were measured at steady state conditions and were used for calculating the overall conductivity of the box.

Eight other experiments were conducted where the cool box was operated at different ambient temperatures with different settings for its internal temperature. In the experiments the measured temperatures were plotted as a function of time. Due to the temperature control method of the cool box the measured temperatures never became constant. Therefore mean values for the measured temperatures and the power input to the FPSC were recorded when visual inspection of the plots of the measured temperatures indicated that the temperatures were oscillating around their steady state levels. The ambient temperature was varied between 25 and 43 ^{o}C. The temperatures at the heat absorber varied between -28.5 and 1.5 ^{o}C and the temperatures at the heat rejector varied between 26.7 and 53.8 ^{o}C. The temperature difference between the heat rejector and the heat absorber therefore varied between 25.7 and 78.6 ^{o}C.

The previously measured overall conductivity was used for computing the rates of heat conduction through the walls of the box in the experiments from the measured temperatures. The thermosyphon, which cooled the vertical inner walls of the cool box, was well insulated between the FPSC and the inner walls of the cool box. It was therefore assumed that the rate of heat flow into the heat absorber of the FPSC, i.e. the cooling power of the FPSC, was equal to the rate of heat conduction through the walls of the cool box.

The model setup

The temperatures measured on the outside of the casing of the FPSC at the locations of the heat rejector and the heat absorber were used directly as boundary conditions for the Stirling machine model in the simulations. The operating frequency and the filling pressure of the FPSC were printed on the label on the outer casing of the FPSC as shown on the photo in Fig. 32. But the strokes of the piston and displacer were not measured in the experiments.

It was chosen to set the strokes of the work piston so that the calculated values for the cooling power agreed with the values determined in the experiments. The comparison of simulation results and experimental data for the FPSC hence had to be done by checking if the model was able to predict the power consumption and thereby the coefficient of performance, COP.

The experimental data contained the time averaged electrical power input to the linear motor of the FPSC, and the *MUSSIM* model calculated the time averaged rates at which the linear motor had to exert work on the work piston. In order to compare simulations and experiments it was therefore necessary to use an approximation for the efficiency of the linear motor of the FPSC. The approximation was made by deriving simple expressions for the losses in the copper winding and the iron of the linear motor.

The copper losses are proportional to the electrical resistance of the winding and to the square of the RMS electric current flowing through the winding.

The electrical resistance of the winding was measured directly on the disassembled FPSC. An equation for the RMS current was established by setting the work output from the motor equal to the product of the motors electromagnetic constant, the RMS current, and the RMS work piston velocity. The electromagnetic constant was estimated from the geometry of the linear motor and a guess for the magnetic properties of the magnet in the linear motor. The RMS work piston velocity was known from the simulations. The RMS current could then be calculated as it was the only unknown quantity.

The iron losses were estimated by multiplying the specific iron losses (W/kg), expressed as a function of magnetic flux density, with the total iron mass in the linear motor.

The magnetic flux density in the iron material was estimated from the geometry of the magnetic circuit of the linear motor and from the RMS current. The estimated flux density contained uncertainties both due to the guess for the magnetic properties of the magnet material and because fringe fields were not considered.

It has been published (Jansson, 2004)(Powder Metallurgy, 2005) that the iron material in the linear motor of the Twinbird FPSC is made from a Somaloy iron powder by Höganäs. The specific iron loss as a function of magnetic flux density was expressed as a polynomial fit of catalogue data for Somaloy 500 iron powder at 80 Hz frequency. The fit introduced some uncertainties because it could actually have been a different material from the Somaloy product family, and because only few data points could be read from the catalogue.

Comparison of experimental data and simulation results

The temperatures at both the compression and expansion ends of the Twinbird FPSC had been varied in the experiments. Therefore it was chosen to plot the results from the comparison using the Carnot coefficient of performance, COP_{Carnot}, as the horizontal axis. The values for COP_{Carnot} were calculated from the measured temperatures at the heat absorber and heat rejector. The results of the comparison are shown in Fig. 33.

In Fig. 33 there is only one data series for the cooling power because the simulations were made to match the measured values for the cooling power by adjusting the strokes of the work piston. The experimental and calculated values for the cooling power were therefore identical.

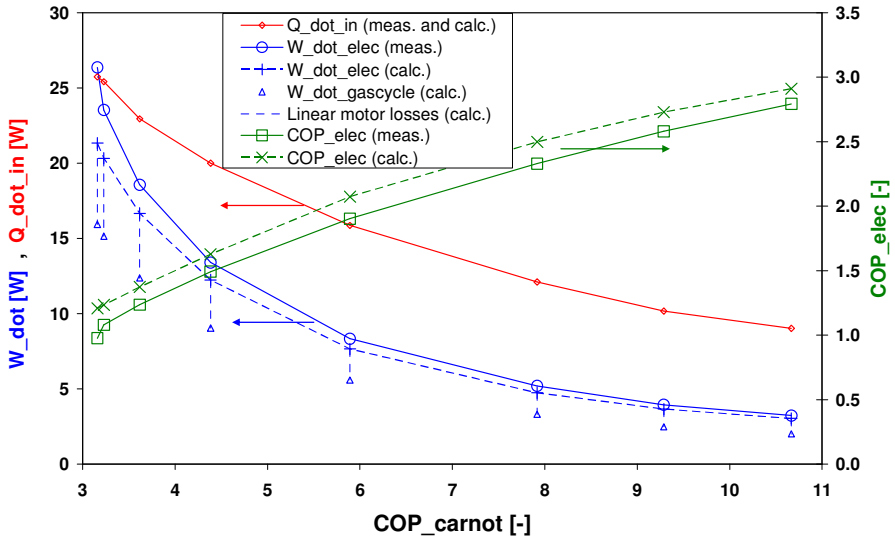


Fig. 33. Comparison of experimental data (solid lines) and simulation results (dashed lines) for the power consumption and COP of the Twinbird FPSC. The calculated values of the work input to the gas cycle are also shown (Δ-symbols) and the vertical dashed lines indicate the power dissipated in the linear motor.

Fig. 33 shows that the *MUSSIM* model, which included the simple expressions for the losses in the linear motor, appeared to slightly overestimate the COP of the Twinbird FPSC. The electric power consumption was underestimated by 6 to 19 %.

The calculated efficiencies of the linear motor varied between 66 and 75 %. The vertical dashed lines in the figure show how much of the input power was dissipated as losses in the linear motor in the simulations. The rates of work input to the work piston from the simulations are also shown in Fig. 33 (Δ-symbols).

Due to the moderate efficiency of the linear motor the uncertainties in the calculations of the losses in the linear motor contributed significantly to the uncertainties in the calculated electrical power consumptions of the FPSC.

The phase angle lead of the displacer relative to the work piston varied between 80 and 88 degrees in the simulations.

The simulations of the Twinbird FPSC required less than 25 % of the CPU time needed for simulations of the SM5 engine with the same model, spatial discretisation, and settings for the IVP solver. This was because the compression ratios and the Reynolds numbers were smaller for the Twinbird FPSC than for the SM5 engine. This caused the equations to be more linear so that fewer steps and fewer iterations pr. solution were needed.

Comparison to simulation results from PROSA

The simulation results from the *MUSSIM* model were also compared to results calculated using the *PROSA* simulation program by Thomas (2006). The linear motor was not included in the comparison with *PROSA*. The results compared below are therefore for the gas cycle only, and hence the calculated values for COP are higher than the values presented in Fig. 33.

A *PROSA* input file for the Twinbird FPSC was created so that the geometry corresponded closely to the geometry modelled by the *MUSSIM* model. This input file was then used with both the current version 2.4 of *PROSA* and a beta version of the new *PROSA 3.0*. *PROSA 2.4* is based on closed form solutions whereas the new version *3.0* uses a time stepping integration to perform an actual simulation of the cycle.

Both the tested versions of *PROSA* use a number of simplifications compared to the *MUSSIM* model.

PROSA uses a lumped formulation for the heat exchangers and just four control volumes for the regenerator. *PROSA* relies on a log mean temperature difference approximation in order to correctly predict the heat transfer. The appendix gap is not modelled directly but the appendix gap losses are taken into account using empirical correlations. Matrix temperature oscillations are also not modelled directly and are also accounted for by correction terms. *PROSA* does not include radial heat conduction.

The simplifications contribute to making even the new version 3.0c of PROSA very fast compared to the *MUSSIM* model. But the simplifications can also affect the accuracy of the simulations. However, the simulations using the *MUSSIM* model indicated that the design of the Twinbird FPSC was such that the simplifications made in *PROSA* should not have a major impact on the calculated performance.

The beta version, *PROSA 3.0c*, did not support simulation of free piston machines, although this will be supported in the release version. With *PROSA 3.0c* the Twinbird FPSC therefore had to be modelled as a crank operated machine by importing the motion of the pistons from the *MUSSIM* model. The comparison with *PROSA 3.0c* was

therefore restricted to be a comparison of the thermodynamics only, as the calculated piston kinematics could not be compared.

Initial simulations showed that there were significant differences between the regenerator losses predicted by the two models. Through correspondence with Thomas (2006) it was found that different assumptions were made in the two models about the Nusselt numbers in the foil type regenerator of the Twinbird FPSC.

In the *MUSSIM* model it was assumed that the Nusselt number for laminar flow between the layers of foil was equal to 8.23. This value corresponds to the analytical solution for incompressible steady state flow between infinite parallel plates where the heat flux is constant in the flow direction. In *PROSA* it was assumed that the Nusselt number for laminar flow between the layers of foil was equal to 3.65. This value corresponds to the Nusselt number for steady, incompressible, laminar flow in a circular tube with a constant wall temperature in the flow direction. Both *PROSA* and the *MUSSIM* model defined the hydraulic diameter as twice the distance between the layers of foil. Therefore the heat transfer coefficients for laminar flow in the regenerator were more than twice as large in the *MUSSIM* model as in the *PROSA* model.

The difference between the assumptions meant that the regenerator losses predicted by *PROSA* were significantly larger than the regenerator losses predicted by the *MUSSIM* model. In order to make a comparison where the large difference in the regenerator loss would not mask other differences between the models, the *MUSSIM* model was temporarily modified so that it used the same assumption as *PROSA* for the Nusselt number for laminar flow in the foil regenerator.

A comparison of results for the following model setups was made:

1. The *MUSSIM* model with the second refinement of the spatial discretisation from Table 1.
2. The *MUSSIM* model with the second refinement of the spatial discretisation, and the Nusselt number for laminar flow in the foil regenerator set equal to 3.65.
3. *PROSA 3.0c* where the FPSC was modelled as a crank operated machine. The motion of the pistons was imported from setup 2.
4. *PROSA 2.4* where the FPSC was modelled as a crank operated machine. The motion of the pistons was again imported from setup 2.
5. *PROSA 2.4* where the FPSC was modelled as a free piston machine.

The 8 operating conditions corresponding to the above mentioned experiments with the cool box were simulated. For the setups 1, 2, and 5 the strokes of the work piston were adjusted so that the cooling powers matched the experiments. The cooling powers from setup 3 and 4 deviated slightly from the experimental values because the *PROSA* models gave slightly different results than the *MUSSIM* model for the same piston motions.

The comparison of results showed that the results predicted by the model setups 2 through 5 were very similar, but that the results calculated using setup 1 had a significantly higher COP. The calculated COPs for the different model setups are compared in Fig. 34.

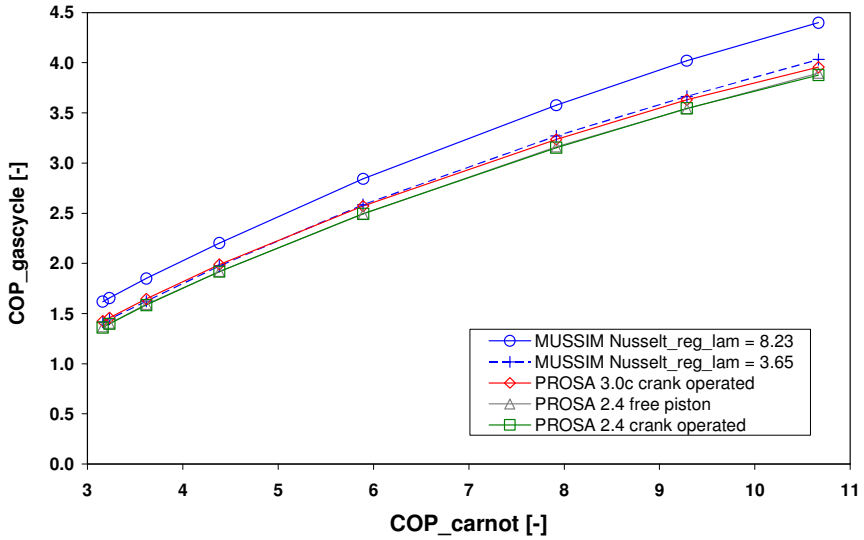


Fig. 34. Comparison of the COP predicted for the gas cycle of the Twinbird FPSC by the *PROSA 2.4*, *PROSA 3.0c*, and the *MUSSIM* model.

Fig. 34 shows that:

- The different assumptions about the Nusselt numbers in the regenerator caused a 9 to 15 % difference in the COP values for the gas cycle.
- The simulations with *PROSA 3.0c* yielded COP values that deviated less than 2 % from the results from the *MUSSIM* model where the Nusselt number for laminar flow in the regenerator was set equal to 3.65.
- *PROSA 2.4* predicted COP values that were 2 to 4.5 % lower than the COP values predicted by *PROSA 3.0c* when the same piston motions were used.
- The COP values predicted by *PROSA 2.4* were almost identical when the FPSC was modelled as a free piston machine and as a crank operated machine with piston motions from the *MUSSIM* model.

For the model setups 2, 3, and 4, which all used the same piston motions, there were some differences between the predicted cooling powers:

- *PROSA 3.0c* predicted cooling powers which were 4 to 7.5 % larger than the cooling powers predicted by the *MUSSIM* model.
- The cooling powers predicted by *PROSA 2.4* were 6 to 9 % smaller than the cooling powers predicted by the *MUSSIM* model.
- The cooling powers predicted by *PROSA 3.0c* were 14 to 15 % larger than the cooling powers predicted by *PROSA 2.4*.

The results indicated that the major difference between the COP values calculated with *PROSA* and with the *MUSSIM* model were due to the different assumptions about the Nusselt numbers for laminar flow in the regenerator. Without this difference the results were very similar; especially for the new *PROSA 3.0* and the *MUSSIM* model.

The results also show that the closed form solutions used in *PROSA 2.4* were quite close to the results of the time stepping solution from *PROSA 3.0c* for this particular Stirling machine design.

There were some differences in the cooling powers predicted by the different models even when the same piston motions were used. But to this it must be added that the cooling power was quite sensitive to the strokes of the pistons.

Uncertainties
The uncertainties in the comparison of the *MUSSIM* model with the Twinbird FPSC were also divided into the following three categories:

- Uncertainties in the experimental results.
- Uncertainties in the thermodynamic analysis.
- Uncertainties in the calculations of the efficiency of the linear motor.

The measurements of the temperatures and the power input to the FPSC were believed to be accurate. The measurements of the cooling power were probably less accurate as the method of using an overall conductivity to determine the cooling power was quite indirect.

One source of uncertainty due to this method was related to the temperature distribution on the inner surfaces of the cool box. In the experiments where the overall conductivity was determined the box was heated internally with a light bulb. In these experiments the temperature distributions on the inner walls of the box were probably reasonably uniform. But the FPSC was connected via a thermosyphon to only the vertical inner walls of the box. So when the FPSC was cooling the box the vertical walls might have been colder than the lid of the cool box. This would presumably cause the need for cooling to be slightly overestimated by the overall conductivity.

The materials and the shape of the FPSC are such that the losses due to heat conduction in the solids of the machine were moderate. This reduced the impact of the simplifications made when modelling the multi-dimensional heat conduction in the FPSC.

By making additional parametric studies it was also found, that the thermodynamic analysis in the *MUSSIM* model was relatively insensitive to moderate changes in the heat transfer and pressure losses predicted by the empirical correlations used in the model. When all calculated friction factors in the model were multiplied by a factor of 1.5 then the calculated values for COP were reduced by only 3 to 10 % with the largest changes for the smallest cooling powers. The sensitivity of COP to the Nusselt numbers for the regenerator was also found to be moderate in the comparison with *PROSA* results.

The comparison with *PROSA* also indicated that the performance prediction for the Twinbird FPSC was relatively insensitive to simplifications which could be made when modelling the FPSC.

Based on these results it was concluded that the uncertainties in the thermodynamic analysis appeared to be small.

The uncertainties in the estimation of the efficiency of the linear motor were discussed above, but the magnitudes of the uncertainties have not been quantified.

Conclusions
The study yielded good agreement between simulation results and experimental data for the COP of the Twinbird FPSC; the Stirling machine model from section 3.4 appeared to underestimate the power consumption of the FPSC by 6 to 19 %. When an important assumption made in *PROSA* about the Nusselt number in the regenerator was duplicated, then close agreement was also found between *PROSA* and the Stirling machine model from section 3.4.

The simulation results were relatively insensitive to small perturbations of the empirical correlations used for calculating friction and heat transfer. A comparative study with a *PROSA* model supported that the thermodynamic analysis of the FPSC was also relatively insensitive to the assumption made when modelling the machine. The major contributors to the observed differences between simulation results and experimental data were therefore assumed to be related to the experiments and the simplified calculation of the losses in the linear motor.

3.9.3. Validation against Sage simulation results for a large Stirling type Pulse Tube cooler

A model of a Stirling type pulse tube cooler (PTC) was created for studying temperature inhomogeneities which had been observed in the regenerator of a large PTC with a 10 kW Pressure Wave Generator (Dietrich, Yang, and Thummes, 2005)(Gromol et al., 2006).

In order to verify that the PTC model was good enough to be usable for the numerical study on temperature inhomogeneities in regenerators the model was validated against results from a model built by Dietrich in the simulation software *Sage*. For convenience the two PTC models are denoted simply as the *MUSSIM model* and the *Sage model* in the remainder of this section.

The *MUSSIM* model was needed for the study on temperature inhomogeneities in regenerators because the existing *Sage* model could not be used for studying the temporal evolution of the inhomogeneities; *Sage* is strictly for computing periodic steady state solutions. The *MUSSIM* model was used to reproduce and study an instability phenomenon which can occur in parallel regenerator channels or in regenerators with wide aspect ratios. After the instability had been reproduced by the *MUSSIM* model the effects of the instability on periodic steady state solutions were also studied using a modified version of the *Sage* model by Dietrich. Results that explain the driving mechanism behind the instability phenomenon which caused the observed temperature inhomogeneities in regenerators were presented in the accompanying *Paper C*.

The *Sage* model had previously been validated against experimental results for the PTC. It had been found that the model was able to predict the cooling power of the PTC with acceptable accuracy. The wave form of the pressure oscillations at different locations in the PTC had also been compared to experimental data. It had been found that the model could reproduce with reasonable accuracy the amplitudes and the phase angles between the pressure waves at different locations in the PTC but that there were some discrepancies between the shapes of the measured and calculated pressure waves.

In the comparison of the models reported on here the models both contained the basic components of a PTC shown in Fig. 5. But there were some additions in order to take into account the geometry of the actual PTC. The heat exchangers at the ends of the pulse tube itself, for instance, were designed so that they had wire mesh sections facing the pulse tube in order to straighten the gas flows blowing into the pulse tube. The designs of the heat exchangers were taken into account in both models by representing the heat exchangers with appropriate serial connections of simple components.

The *Sage* model included a detailed description of the dual opposing piston pressure wave generator whereas the volume variation in the pressure wave generator was prescribed explicitly in the *MUSSIM* model. In the *MUSSIM* model simplifications

relative to the *Sage* model were also made at the locations where the inertance tube was connected to its neighbouring components. Except for these differences, however, the models were close to identical with respect to the modelled geometry.

Heat conduction in the solids of the components in the PTC were included in both models. But the conduction paths between the different components were not included in the *MUSSIM* model. To get from one component to the next the heat conducted inside the walls of components hence had to get absorbed and transported by the gas in the PTC.

The *Sage* model used empirical correlations developed by Gedeon and Wood (1996) for friction and heat transfer in the regenerator. The *MUSSIM* model used regenerator correlations of the form suggested by Thomas (2000) which had been fitted to match the correlations by Gedeon and Wood. The *Sage* model also used empirical correlations to account for multi-dimensional flow patterns in the pulse tube. These correlations had only negligible influence on the results of the *Sage* model and were not included in the *MUSSIM* model.

It should be noted that these multi-dimensional flow patterns in the pulse tube are likely to be one of the major sources of uncertainties for the one-dimensional PTC models. The actual flow patterns inside a pulse tube could depend strongly on inhomogeneities in the gas flows into the pulse tube. Hence some PTC designs could render the predictions of empirical correlations for the associated energy transport inaccurate.

The two models were compared at an operating frequency of 50 Hz and a mean pressure in the pressure wave generator of approximately 2 MPa. Heat was absorbed at 60 K and was rejected at 300 K. The volume oscillations in the pressure wave generator found by the *Sage* model were prescribed explicitely in the *MUSSIM* model, so that the volume variations were identical in the two models.

Comparison of performance numbers

For the two models the following performance numbers were collected and compared:

- The rate of work input by the pressure wave generator to the gas.
- The cooling power of the PTC.
- The net flux of heat pumped against the axial temperature gradient in the gas in the pulse tube.
- The net axial enthalpy flux carried by the gas though the regenerator.

The results of the comparison are shown in Table 7.

	Sage model	MUSSIM model	Rel. difference [%]
Work input [kW]	7.06	6.88	-2.5
Cooling power [W]	230	225	-2.2
Pulse tube heat pumping [W]	499	484	-3.1
Regenerator loss [W]	245	260	6.1

Table 7. Comparison of the rates of work input, the cooling powers, heat fluxes pumped in the pulse tube, and the regenerator losses predicted by the *MUSSIM* model and the *Sage* model.

Table 7 shows that the results from the two models agreed to within a few percent. The differences were of the same order of magnitude as the expected errors due to the coarsenesses of the discretisations used in the models. This indicates that the differences between the models appear to have only minor impact on the performance prediction of the two models.

Comparison of pressure waves

Fig. 35 shows a comparison of the pressure waves predicted by the two models for different locations in the PTC.

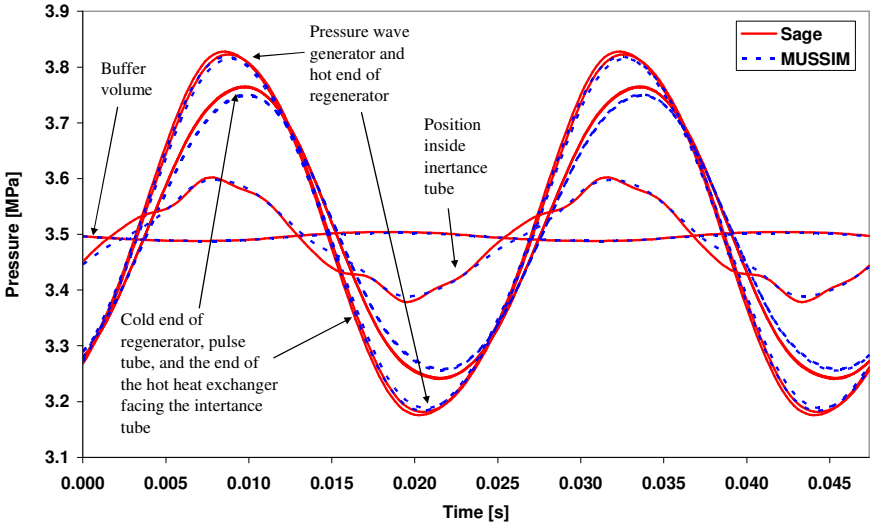


Fig. 35. Comparison of pressure waves predicted by the *MUSSIM* model and the *Sage* model. The pressure waves are depicted for two full cycles at 50 Hz operating frequency.

Fig. 35 shows that the shapes and phase angles of the pressure waves predicted by the models were very similar. But some differences between the pressure waves were visible.

The most obvious difference was that the *Sage* model predicted higher pressures in the buffer volume than the *MUSSIM* model. The shapes and phase angles of the pressure waves in the buffer volume were almost identical but *Sage* predicted a mean pressure which was approximately 23 *kPa* higher than in the results of the *MUSSIM* model.

The differences between the mean pressures in the buffer volume were attributed to the differences between the ways that the inertance tube was connected to its neighbouring components in the two models. In the *Sage* model the mean pressure in the buffer volume was approximately 23 *kPa* higher than the mean pressure at the end of the hot heat exchanger facing the inertance tube; the *MUSSIM* model predicted almost no difference in mean pressures between these two locations.

There were also minor differences in the calculated pressure differences between the ends of the regenerator. The differences were attributed partly to inaccuracies in the recalculations of the regenerator correlations and partly to a difference between the assumptions made about flow velocities in the regenerator.

In *Sage* the average flow velocities in the regenerator were calculated from the assumption that the effective flow area in the regenerator was given by the product of the flow area of the empty regenerator canister and the porosity of the regenerator matrix. This assumption was shared by the *MUSSIM* model for foil type regenerators but not for regenerators made from wire screens, such as the regenerator in the PTC model. In the *MUSSIM* model the effective flow area in wire screen matrices was assumed to be given by the area of the "holes between the wires" in one of the wire screens. This assumption lead to higher average flow velocities in the *MUSSIM* model and hence to momentum being more important. Note that the differences in flow areas

were taken into account when the regenerator correlations by Gedeon were recalculated for the *MUSSIM* model.

Conclusions

The performance data and pressure waves calculated with the *MUSSIM* model agreed very well with the corresponding results calculated with the experimentally validated *Sage* model.

Both models solved discrete approximations to the governing equations for one-dimensional compressible fluid flow on Eulerian grids. There were differences between the discretisations of the governing equations used in the Sage model (differential form) and the MUSSIM model (integral form) and there were differences between the assumptions made about the one-dimensional flow. But the results presented above indicate that the differences between the models did not make a big difference for this particular PTC design.

3.9.4. Validation against Sage simulation results for the AIM Pulse Tube cooler

The PTC model described in section 3.9.3 was further refined for a study on radial heat conduction in a small PTC prototype developed by the company AIM INFRAROT-MODULE GmbH for infrared applications. The AIM pulse tube cooler had a compact coaxial design where the pulse tube was placed inside the regenerator. The capabilities of the commercial AIM PTCs have been described in detail by Korf et al. (2005). The AIM pulse tube cooler is shown in Fig. 36.

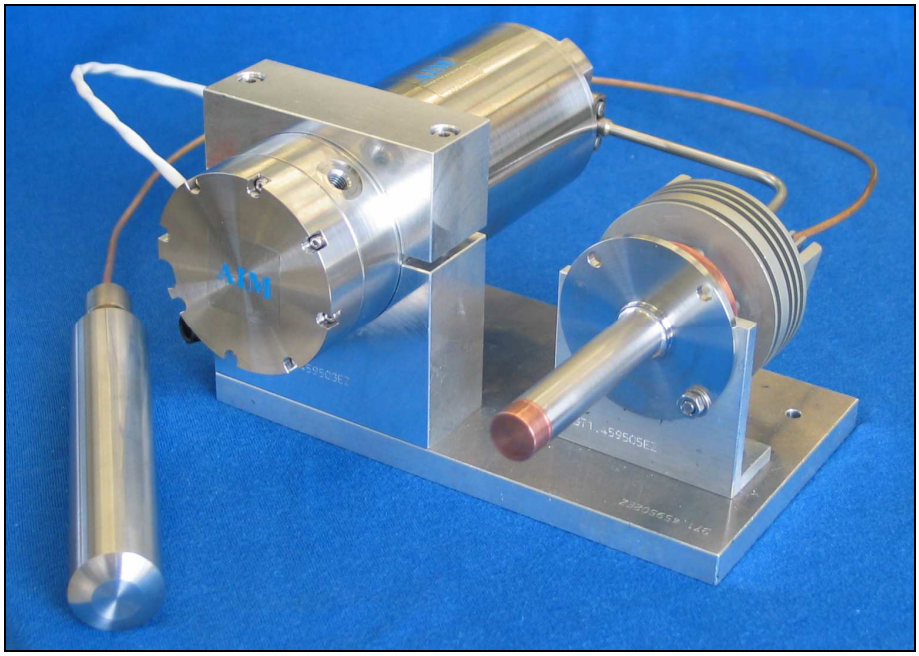


Fig. 36. Coaxial AIM pulse tube cooler designed for infrared applications.

In Fig. 36 the largest cylindrical object is the pressure wave generator. The pressure wave generator is connected via a transfer line to the so called pulse tube *cold head* in the right hand side of the image. Despite its name the cold head actually has both a warm end and a cold end, and the cold head contains both the after cooler, regenerator, cold heat exchanger, pulse tube, and hot heat exchanger (see Fig. 5). The narrow

cylindrical protrusion on the cold head is known as *the cold finger*. It is the tip of the cold finger which is used for cooling infrared sensors. The regenerator and the pulse tube are placed inside the cold finger. The warm end of the cold head is positioned at the opposite end from the cold tip, and this is where the after cooler and the hot heat exchanger are located. The cold head is connected via the inertance tube to the buffer volume, which is shown in the lower left corner of Fig. 36.

The refined PTC model was validated against a *Sage* model, which was being developed at AIM. In the remainder of this section the two models of the AIM PTC are denoted as the *MUSSIM model* and the *Sage model*.

The *Sage* model of the AIM PTC included a detailed description of the pressure wave generator whereas the volume variation in the pressure wave generator was prescribed explicitly in the *MUSSIM* model. Except for differences in the internal geometry of the pressure wave generator there were no significant differences between the geometries defined in the two models.

The radial heat conduction in the cold head was not included in the *Sage* model. The radial heat conduction in the cold finger in the *MUSSIM* model was disabled during the comparison of the models. Axial heat conduction in and between components were included in both models. The models used the same empirical correlations for flow friction and heat transfer as were used in the comparison with the large PTC described in section 3.9.3.

For the AIM PTC, however, the correlations used in the *Sage* model to account for axial heat transport due to multi-dimensional flow patterns in the pulse tube did have a significant impact on the simulation results. *Sage* applied the corrections for the multi-dimensional effects in the pulse tube as a multiplier to the molecular heat conduction in the pulse tube. *Sage* also produced estimates for the net contributions to the axial heat transport in the pulse tube. According to these estimates the multi-dimensional effects in the pulse tube caused an axial energy transport which was approximately 70 times as large as the energy transport due to molecular heat conduction. In order to compensate for the absence of this energy transport due to multi-dimensional flow patterns in the *MUSSIM* model the rates of molecular heat conduction in the pulse tube were multiplied by a factor of 70 in the *MUSSIM* model. Note that this quick fix for the missing loss terms in the *MUSSIM* model is in no way general; the fix is specific to both the operating conditions and geometry of the PTC.

The two models were compared at an operating frequency of 42 Hz and a mean pressure in the pressure wave generator of nearly 3.5 MPa. Heat was absorbed at 75 K and was rejected at 300 K. The volume oscillations in the pressure wave generator found by the *Sage* model for a specific rate of work input to the gas by the pressure wave generator were prescribed directly in the *MUSSIM* model.

Comparison of performance numbers

For the comparison of the two models of the AIM PTC the same performance numbers as in section 3.9.3 were studied, namely:

- The rate of work input by the pressure wave generator to the gas.
- The cooling power of the PTC.
- The net flux of heat pumped against the axial temperature gradient in the gas in the pulse tube.
- The net axial enthalpy flux carried by the gas though the regenerator.

The results of the comparison are shown in Table 8.

	Sage model	MUSSIM model	Rel. difference [%]
Work input [W]	37.2	35.8	-3.8
Cooling power [W]	1.40	1.19	-14.9
Pulse tube heat pumping [W]	4.45	4.33	-2.8
Regenerator loss [W]	2.53	2.85	12.6

Table 8. Comparison of the rates of work input, the cooling powers, the rates of heat pumping by the pulse tube, and the regenerator losses predicted by the two models of the AIM PTC.

Table 8 shows that the results from the two models agreed quite well, but that there were some differences. The *Sage* model predicted slightly better heat pumping in the pulse tube, a smaller regenerator loss, and, hence, a better cooling power. The cooling powers predicted by the models are relatively close to experimental data for the AIM PTC published by Korf et al. (2005).

The larger regenerator losses in the *MUSSIM* model were attributed to the slightly different regenerator correlations used in the models, as described in section 3.9.3.

The constant multiplication factor applied to the molecular heat conduction in the *MUSSIM* model, in order to compensate for the lack of correlations for multi-dimensional flow effects in the pulse tube, presumably contributed to the differences in the heat pumping of the pulse tube.

The coarseness of the discretisations used in the models might also have contributed to the differences.

Comparison of pressure waves

Fig. 37 shows a comparison of the pressure waves predicted by the two models for different locations in the PTC.

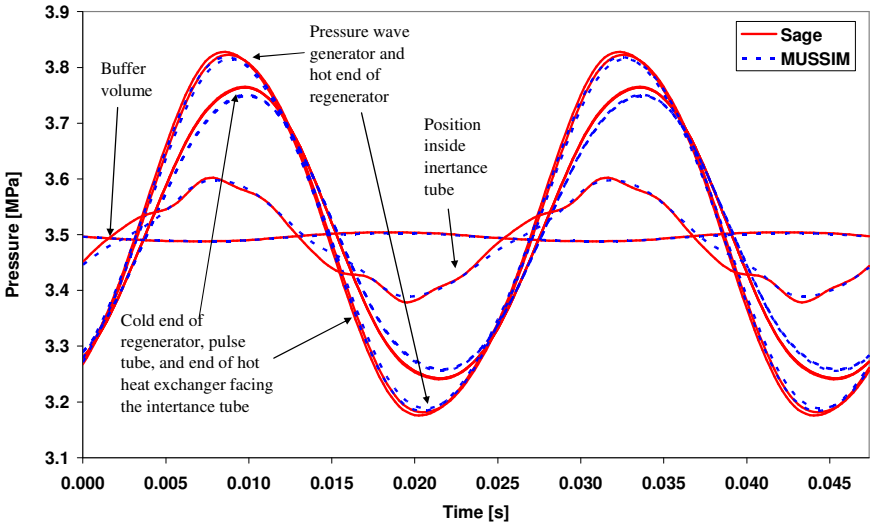


Fig. 37. Comparison of pressure waves predicted by the two models at different locations in the AIM PTC. The pressure waves are depicted for two full cycles at 42 Hz operating frequency.

Fig. 37 shows that the pressure waves predicted by the models at different locations in the PTC were very similar with respect to both amplitudes and phase angles, but that there were some differences.

The pressure waves in the buffer volume were very similar for the two models but there were some differences in the pressure losses across the inertance tube. The models both simulated a composite inertance tube which consisted of several tube sections with different diameters.

Fig. 37 shows that at a position inside the inertance tube the pressure waves agreed quite well although the pressure wave predicted by the *Sage* model is somewhat more irregular than the pressure wave predicted by the *MUSSIM* model. A contribution to the irregularity could be due to the *Sage* model approximating a nearly triangular wave form with only eight harmonics due to the chosen temporal discretisation in the *Sage* model.

Fig. 37 also shows that there were minor differences in the magnitudes of the total pressure drops predicted across the composite inertance tube. A possible explanation for this is that the localised losses at tube inlets and outlets were handled differently in the two models when the simulations for the comparison were made. In the *MUSSIM* model a local loss coefficient of 0.5 was applied when the gas flow went through a reduction in the available flow area. When the gas flow went through an expansion in the flow area a local loss coefficient of 1.0 was applied. The local loss coefficients were applied locally at the flow area discontinuities. In the *Sage* model the pressure losses due to the local loss coefficients were distributed evenly over the entire lengths of the components. Each of the components, which represented a section of the inertance tube with a specific diameter, were prescribed a loss coefficient of 1.5 corresponding to the sum of the local loss coefficients for one inlet and one outlet. At the locations where two inertance tube sections with different diameters were connected this practice resulted in each flow area discontinuity contributing with loss coefficients for both an inlet and an outlet.

It also appears that the pressure losses in the regenerator were predicted as slightly larger by the *MUSSIM* model than by the *Sage* model. The reasons for the slightly larger pressure losses in the regenerator were assumed to be the same as were discussed in section 3.9.3.

The differences in the pressure waves must have also contributed to the differences between the performance numbers predicted by the two models.

Conclusions
The agreement between the two models of the AIM PTC was good. The cooling powers predicted by the models were close to published experimental data for the AIM PTC.

3.9.5. Conclusions

It was found that the performance predicted for the SM5 Stirling engine by the Stirling machine model described in section 3.4 agreed well with experimental data over a wide range of operating conditions.

The Stirling machine model was also found to predict the performance of a free piston Stirling cooler by Twinbird well. When an important assumption about the heat transfer in the regenerator was matched to the assumption made in *PROSA* then the results from the Stirling machine agreed very well with simulation results from *PROSA*.

In a comparison of a model of a large pulse tube cooler with an experimentally validated *Sage* model it was found that there were close agreement between the two models. Both the predicted performance numbers and the predicted pressure waves at different locations inside the PTC were found to agree closely.

Comparison of two different models of a small pulse tube cooler prototype, which had been developed by the company AIM for infrared applications, also showed good agreement between a *Sage* model and a model built using the modelling approach described above. The cooling powers predicted by the models were also close to published experimental data for the AIM pulse tube coolers.

It is therefore concluded that the one-dimensional modelling approach presented above is sufficient to produce results in good agreement with experimental data for both Stirling machines and pulse tube coolers. It was also found to produce results in good agreement with the current state of the art software, *Sage*.

Pulse tube coolers, however, stretch the assumptions of the one-dimensional models further than Stirling machines do. This is because multi-dimensional flow patterns in the pulse tube, which are difficult to accurately account for in a one-dimensional model, can have significant impact on the performance of the coolers.

3.10. Summary

The theory behind a control volume based approach for modelling oscillating, compressible flow in one space dimension has been presented. The approach chosen in this work for implementing the modelling approach has also been described. The necessary conditions for solutions to models to be unique, periodic steady state solutions have been stated.

The chosen approaches produce models where all the equations, which are on a form that should be understandable to someone with a background in engineering thermodynamics, can be accessed and modified individually. The implementation was designed to be easy to work with when creating and modifying models and to make simulations fast.

Comprehensive testing of both the implementation and the intrinsic properties of the modelling approach has been performed.

The results from the tests of the implementation showed that a high level of accuracy could be achieved for integrations of a model created using the modelling approach. It was also found that the level of accuracy depended on the settings for the numerical solvers in a very predictable way. The tests also showed that the choices of numerical methods made a significant difference to the speed of simulations. By making appropriate choices of numerical methods for a given model and a given fineness of the discretisation used in that model, one can ensure that simulations are performed with good computational performance.

The tests of the modelling approach showed that the modelling approach is convergent, i.e. that solutions converge onto the correct solutions for one-dimensional compressible flow when the discretisation is refined. This was found to be the case even when discontinuities and shocks were present in the flow fields. It was also found that all aspects of the modelling approach are conservative. This means that checks for conservation of mass and energy can be used for checking that models have been

constructed correctly, and for checking if solutions are true periodic steady state solutions.

Models created using the modelling approach described here have been validated against both experimental data and against simulation results calculated with the simulation packages *Sage* and *PROSA*. The validation was performed for both a Stirling machine model and two pulse tube cooler models.

It was found that the modelling approach presented here produced simulation results in good agreement with both experimental data for Stirling machines and pulse tube coolers and with simulations results from *Sage*. When an important assumption made in *PROSA* was duplicated the agreement with *PROSA* simulation results was also very good.

4. Numerical methods for finding periodic steady state solutions

This chapter is on the theory, implementation and testing of a numerical method for finding periodic steady state solutions to models built using the modelling approach presented in chapter 3.

This chapter begins with a definition of the numerical problem which must be solved to find periodic steady state solutions. This is followed by a discussion of different methods for solving such problems. In this discussion it is explained why shooting methods were found to be suitable.

The following sections explains the theory of a single shooting method and a multiple shooting method for finding periodic steady state solutions. A method for reducing the dimension of the BVP which must be solved by shooting is discussed along with other methods for reducing the computational efforts needed to find solutions. It is also discussed how the shooting methods have been parallelised.

The section on the theory of the single and multiple shooting methods is followed by sections which describe the tests performed to compare the shooting methods.

The next section is on the additional potential for parallelisation in batch jobs for the shooting methods. The chosen parallelisation approach is described and it is described how the implementation handles failed simulations. The possibility of letting the shooting methods manage the tolerance used in IVP solver based on the current accuracy of the shooting solution is also discussed. Numerical results are presented for the parallel batch method and the results are compared to the performance which was achieved without the specialised batch handling method.

4.1. Problem definition

The problem is to solve a BVP for a system of first order ODEs (or a system of DAEs) which can consist of both periodic variables, integral conditions, and a scaling condition. The solution must satisfy the conditions (17) described in section 3.2.8. Eq. (17) is repeated below for convenience:

$$\begin{aligned} \underline{0} &= \underline{y_{p,0}} - \underline{y_p}\left(t_0 + \Delta t; \underline{y_{p,0}}, \underline{c_i}\right) \\ \underline{0} &= \underline{y_i}\left(t_0 + \Delta t; \underline{y_{p,0}}, \underline{c_i}\right) \\ 0 &= SCTV - y_s\left(t_0 + \Delta t; \underline{y_{p,0}}, \underline{c_i}\right) \end{aligned} \tag{17}$$

A model built using the modelling approach presented in chapter 3 needs only contain ODEs. But one can imagine situations where algebraic equations are convenient or necessary for computing the right hand sides of the ODEs, i.e. where models will actually be DAEs. Consider systems of DAEs where:

- The states in the model can be determined directly from $\underline{y_p}$ and $\underline{c_i}$.
- The algebraic equations have a unique solution for a given set of values in $\underline{y_p}$, and $\underline{c_i}$.

If $\underline{y_p}$ satisfies (17) then the states in the model will vary in a periodic fashion and the model will be in the same state at the beginning and the end of a periodic cycle. The algebraic variables will then automatically also satisfy periodic boundary conditions.

In the remainder of this chapter we consider only systems of ODEs. But the analysis should be extensible to DAEs of the type outlined here, where the algebraic equations can be considered as nothing more than an intermediary step in the evaluation of the right hand sides of the ODEs.

As mentioned in section 3.2.8 the scaling condition is necessary in order to get a unique solution which corresponds to one specific amount of gas contained inside the gas domain of a model. The scaling condition can be formulated either explicitly in the mass of gas, or in a mean value for the pressure at a given location in the gas domain.

For a given model one group of periodic variables, such as the pressures or energy densities, $\rho \cdot e$, in the control volumes of the gas domain, can be expected to be more or less directly proportional to the mean pressure. Another group of periodic variables, such as the specific energies, e, or oscillating regenerator matrix temperatures, can be expected to depend only weakly on the mean pressure.

Consider a scenario where we have a solution which satisfies (17) at, say, 10 MPa mean pressure. If we wish to compute a guess for a new solution with a similar temperature distribution at 11 MPa mean pressure then it makes sense to distinguish between the two groups of periodic variables. We could make a guess for the new solution by increasing the values of the periodic variables which are directly proportional to the mean pressure by 10 % and leave the remaining periodic variables and the wall temperatures unchanged. The scaling condition target value, $SCTV$, and the ODE for the scaling condition can be chosen so that $SCTV \Big/ y_s\left(t_0 + \Delta t; \underline{y_{p,0}}, \underline{c_i}\right) = 1.1$ for this case.

In the following treatment we collect all the periodic variables in the vector $\underline{y_p}$ of length N_p. We consider $\underline{y_p}$ to be the concatenation of the two vectors $\underline{y_p^s}$ and $\underline{y_p^{ns}}$, where $\underline{y_p^s}$ contains the periodic variables which scale with the mean pressure and where $\underline{y_p^{ns}}$ contains the periodic variables which depend only weakly on the mean pressure. We denote the integration variables in the integral conditions for the wall temperatures as the vector $\underline{y_i}$ of length N_i. Finally, we denote the integration variable for the scaling condition as y_s. The total number of dynamic variables in the model, N_{dyn}, is then equal to $N_p + N_i + 1$.

To solve the BVP we must find the N_p initial values, $\underline{y_{p,0}}$, and the N_i parameters, $\underline{c_i}$, so that the corresponding solution to the IVP satisfies the $N_p + N_i + 1$ equations in (17).

4.2. Numerical methods for finding periodic steady state solutions

For systems of ODEs, such as the ones produced by the modelling approach presented in chapter 3, there are several numerical methods which could be considered for finding periodic steady state solutions. Some of these methods are discussed below.

4.2.1. Integration to convergence, convergence acceleration, and extrapolation methods

One approach could be to integrate the equation system forward in time using an IVP method so that the solution evolves towards the periodic steady state solution. This approach will be denoted as *integration to convergence* or *fixed point iteration* from hereon. This approach can be applied to $\underline{y_p}$ in a model.

In the methods most basic form the final values from one cycle become the initial values of the next cycle. The approach can be enhanced by using the scaling condition to scale the final values of $\underline{y_p^s}$ so that $\underline{y_p}$ will approach the solution with the correct mean pressure. We can think of the integration and the scaling towards the mean pressure as a black box function. The input to this function is the initial values and the output is the scaled final values. We iterate the function to find a fixed point, i.e. a point where the output from the function is equal to the input to the function.

Using such a method for finding periodic steady state solutions has the advantage that any issues with discontinuous derivatives can be dealt with as events by the IVP solver used for integrating the equation system inside the black box function.

One problem with integration to convergence is that solutions to models where the regenerator matrix temperatures are modelled as being dynamic, for instance using the lumped formulation (15), will evolve slowly and asymptotically towards periodic steady state. The simulation of several minutes running time for the simulated machine may be required in order for $\underline{y_p}$ to get within the desired tolerance from a true periodic steady state solution. This can mean that a sequence of several thousand revolutions of the machine must be simulated in order to compute a single periodic steady state solution.

Convergence acceleration

The problem with the slow convergence was addressed by Kühl (1990) by using a specialised method to accelerate the convergence of the slowly evolving regenerator matrix temperatures towards their periodic steady state values. The remaining variables, which should converge much faster, were found by fixed point iteration.

The basis for the approach by Kühl is that at periodic steady state the cycle averaged rate of axial energy transport through a cross section of the regenerator should be independent of the axial position within the regenerator if there is no conduction from the outer walls of the regenerator canister. The approach by Kühl corrects the temperatures of the regenerator nodes so that the cycle averaged rates of energy transport between the nodes become more similar.

Extrapolation methods

Extrapolation methods, such as the ε-algorithm discussed by Skelboe (1977, 1980), are a more general approach for finding periodic steady state solutions to systems of ODEs. The ε-algorithm is a method for computing the limit of a series with exponentially decaying terms. The ε-algorithm could be applied either to all elements of $\underline{y_p}$ or just to the slowly evolving elements such as the regenerator temperatures.

Each iteration of the ε-algorithm requires a few consecutive cycles to be simulated in the same way as consecutive cycles would be simulated for the integration to convergence method. The values of $\underline{y_{p,0}}$ after each cycle are stored. The ε-algorithm

then tries to extrapolate from the stored values to the value of $\underline{y_{p,0}}$ when all exponentially decaying terms have died out.

The ε-algorithm is not difficult to implement and it has been tested with some success for accelerating the convergence of $\underline{y_p}$ for the Stirling machine model with a coarse spatial discretisation. The algorithm required on the order of 10 to 20 iterations to find values of $\underline{y_{p,0}}$ which satisfied the periodic boundary conditions in (17) with good accuracy. For the tested implementation of the ε-algorithm each iteration required the simulation of four cycles.

Fulfilling the integral conditions

The integration to convergence approach with or without convergence acceleration or extrapolation is not trivial to apply to models built with the modelling approach presented above. The main difficulty with respect to finding periodic steady state solutions is to find $\underline{c_i}$ so that the integral conditions are fulfilled. The wall temperatures in $\underline{c_i}$ depend on the heat transfer between the surface segments and the working gas, and the wall temperatures can be strongly coupled due to the heat conduction between the control masses.

An intuitive approach to iterating for $\underline{c_i}$ is to divide the net energy accumulation in each control mass during a cycle with the heat capacities of the respective control masses. This approach produces estimates for the changes of the temperatures of the control masses during each cycle. These temperature changes could then be added to the current values of the wall temperatures in order to produce a new guess for the wall temperatures.

This approach has some similarities to solving a coupled system of ODEs from a lumped formulation for the wall temperatures using an explicit Euler method with a step size equal to the cycle period Δt. Unfortunately, this approach can be unstable.

If any wall temperatures are not coupled then one can also think of the difficulties with achieving convergence in terms of the condition for convergence in fixed point iteration. If the absolute value of the derivative of the output from the black box function with respect to the input is too large (larger than 1), then the fixed point iteration will not converge.

Commiso (1994) concluded that in order to have a general method for fulfilling integral conditions in Stirling machine models one would need to wrap the integration to convergence method in an outer iteration loop that solves the integral conditions. In every iteration of the outer loop one would need to compute at least one periodic steady state solution corresponding to the current guess for $\underline{c_i}$ using the integration to convergence method. If the outer iteration loop is a Newton-Raphson type solver, which requires a Jacobian matrix for the integral conditions, then $N_i + 1$ periodic steady state solutions would have to be computed in order to compute a Jacobian matrix by one-sided numerical differencing. The integral conditions would then increase the cost of finding a periodic steady state solution by a factor of at least N_i.

Other methods, which require less computational effort than the approach suggested by Commiso (1994), but are still stable enough for practical use, could possibly be developed for fulfilling the integral conditions.

One such approach, which is similar to the approach suggested by Commiso, is to compute a Jacobian matrix for the integral conditions from transient cycles. Such a Jacobian matrix can then be used after each transient cycle to solve for the changes in wall temperatures which would eliminate the residuals in the integral conditions produced by the transient cycle. When the equation system has sufficiently benign properties the Jacobian matrix could be recycled for many or all of the transient simulations needed to find a periodic steady state solution using integration to convergence (with or without acceleration towards periodic steady state). This approach reduces the cost of computing the Jacobian for the integral conditions from finding $N_i + 1$ periodic steady state solutions to simulating $N_i + 1$ cycles.

It is also possible that an approach which does not require a Jacobian matrix for the integral conditions could be developed. The approach would have to remain stable even for strongly coupled wall temperatures, such as the wall temperatures in an axially discretised heat exchanger made from copper. Also the approach would have to be general enough to handle, for instance, regenerator wall segments which interact both convectively with the gas, through conduction with a regenerator matrix with oscillating temperatures, and with the neighbouring wall segments. Such a method could be very attractive for finding periodic steady state solutions to models with many integral conditions where the efforts required to find periodic steady state solutions with other methods would be unacceptable. No attempts at deriving or implementing such an approach have been made in this work.

Another alternative would be to abandon the use of integral conditions for the wall temperatures altogether so that the ε-algorithm could be used for finding periodic steady state solutions. This approach has also not been pursued in this work.

4.2.2. Finite difference methods

Finite difference methods can also be applied for finding solutions to the boundary value problem (17). The general reference for what follows is Commiso (1994).

To apply the finite difference method the cycle would be discretised on a mesh with a number of time points. Each mesh point would correspond to one set of values for $\underline{y}$. The derivatives of the ODEs at the mesh points would be approximated by finite differences involving the values of $\underline{y}$ at the neighbouring mesh points. For the periodic variables one could choose to apply a finite difference method similar to the method used in *Sage* by Gedeon (1994) which approximates the solution with harmonics.

The complete finite difference equation system could be formulated so that it included the conditions (17) for a periodic steady state solution. The approach would thus produce one large system of non-linear equations. The number of equations would be approximately proportional to N_{dyn} and to the number of time points in the finite difference mesh. When the large system of finite difference equations was solved it would produce the complete solution to the problem including $\underline{c_i}$ and the values of $\underline{y}$ at all the mesh points.

An advantage of using a finite difference method would be that stability could be obtained for a very coarse temporal discretisation (Commiso, 1994). This would enable fast computation of coarse periodic steady state solutions. This advantage was one of

the original motivations for using such a method in the *GLIMPS* code by Gedeon (1986).

Dealing with discontinuous derivatives when using the finite difference method is not trivial. Ideally one should place mesh points at each discontinuity and use one sided differences on both sides of each discontinuity. If the positions of the discontinuities in the cycle changed during the iterations on the finite difference equation system then the mesh should be changed to reflect this. This would be the case, for instance, when the Stirling machine model from section 3.4 was used for modelling free piston machines such as the Twinbird Stirling cooler. In this case it would be the motion of the displacer relative to the cylinder wall which would determine the locations of the discontinuities. The motion of the displacer would not be precisely known until the solution had converged.

Due to the difficulties in dealing with discontinuous derivatives Commiso (1994) chose to ignore their presence and just used equidistant mesh points when using the finite difference method.

When discontinuous derivatives are present it could sometimes be completely acceptable to ignore them if the errors due to the coarseness of the spatial discretisation dominate over the integration errors anyway. This, of course, requires that convergence in the finite difference equation system can be achieved in spite of the unhandled discontinuities.

Some models may not contain any significant discontinuities in the derivatives of the equations. In such cases finite difference methods appear to be attractive.

The finite difference method could become very computationally intensive if highly accurate solutions are needed. When a fine spatial discretisation and a large number of temporal mesh points are needed then the non-linear system of finite difference equations becomes very large.

The periodic steady state solution found to the 4th refinement from Table 1 of the spatial discretisation for the Stirling machine model, for example, had 2704 ODEs and integration of one periodic steady state cycle required approximately 3300 time steps. If we assume that a similar discretisation was used with a finite difference method then the finite difference equation system would contain on the order of magnitude 9 million non-linear equations. A quick estimate indicated that the Jacobian matrix for such a non-linear system of equations would contain more than 560 million non-zero entries if the discretisation was made using a 4th order divided differences approach. Using a sparse matrix technique each element would typically require 12 bytes of storage. The (non-LU-decomposed) Jacobian would hence require roughly 6.5 GB of storage. It appears to be a daunting task to LU-decompose such a matrix on a present day personal computer.

4.2.3. Shooting methods

The basic idea of the *single shooting* method is to solve the equations for the boundary conditions just as one would solve any other system of non-linear residual equations.

The solution to the equation system is a set of initial values, $\underline{y_{p,0}}$, and a set of parameter values $\underline{c_i}$. Evaluating the residuals of the equation system for the boundary conditions requires one entire cycle to be integrated. An iterative equation solver will be

applied to find a set of values for $\underline{y_{p,0}}$ and $\underline{c_i}$ so that the final values of the integration process result in zero residuals for the equation system.

If one imagines plotting curves for the integration variables while the IVP solver progresses through the integration interval during the solution process, then it is not hard to imagine how the method got its name. An imaginary example of such a set of "tracer" curves generated by a single shooting method is shown in Fig. 38. The shooting method will continue to shoot until it has a solution corresponding to "a perfect broadside" where all integration variables simultaneously hit their targets at the end of the integration interval.

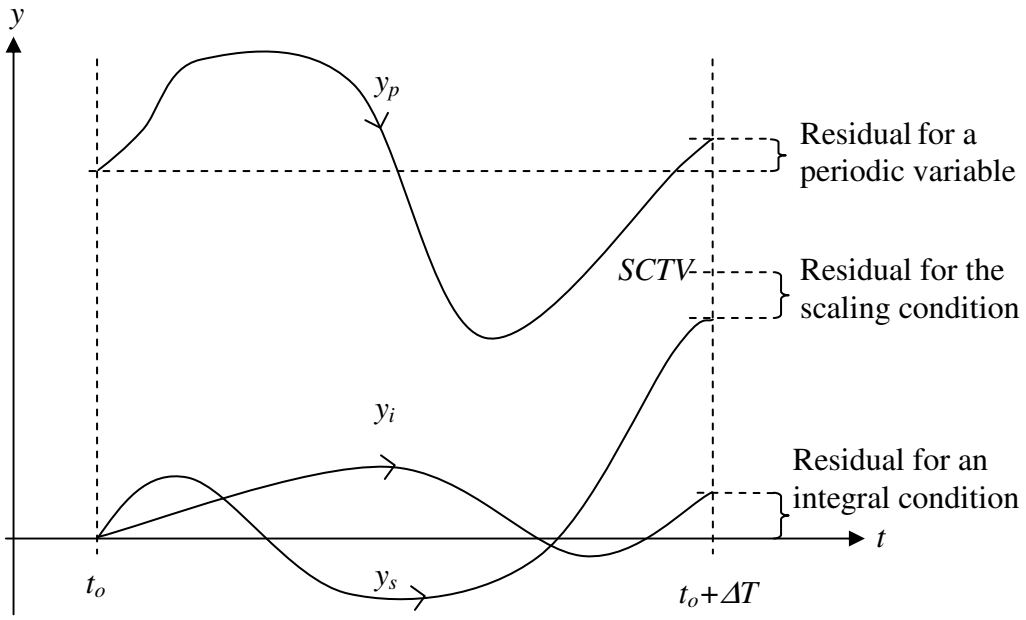


Fig. 38. Illustration of the single shooting method.

One cannot use a standard Newton-Raphson solver to solve (17) because the number of equations is one higher than the number of variables which we solve for. The additional equation can be dealt with by introducing an additional variable, a scaling factor, and this will be discussed below.

For some types of boundary value problems the shooting method can be unreliable. This is the case for problems where the integrations of the cycle can easily fail if the initial values and parameter values lead to an unfortunate integration path.

One example of such a problem could be diffusion of gaseous components into a catalyst pill where some of the components are consumed. The concentration distributions of the components could be computed by integrating a suitable system of ODEs from the surface of the pill towards the centre of the pill. To find the steady state concentration distributions one could try to guess the correct concentration gradients at the surface of the catalyst pill which would lead to vanishing gradients at the centre of the pill. But if the shooting method makes a guess with too steep a gradient at the surface of the pill then a concentration could well become negative during the integration process even though negative concentrations are physically impossible. Negative concentrations could make it impossible to compute the right hand sides of the ODEs and thereby make an integration fail (Andersen, 2003).

Such problems, however, are not relevant for models built using the modelling approach presented above. The properties of the equation systems for the models are such that unfortunate initial values may cause fast transients and different types of shocks and discontinuities. But the fast transients and shocks will dissipate and will not make integrations impossible.

A number of variants of shooting methods exist. In multiple shooting methods the integration interval is cut into multiple sub-intervals. The splitting of the cycle into sub-intervals means that additional equations for the continuity of the solution across the cuts are needed. The concept of multiple shooting is illustrated in Fig. 39. Both single and multiple shooting methods can be applied for finding periodic steady state solutions.

Some variants of multiple shooting integrate in different directions in different sub-intervals. But the properties of the models considered here are such that solutions are unstable if integrated backwards in time and hence this is not an option.

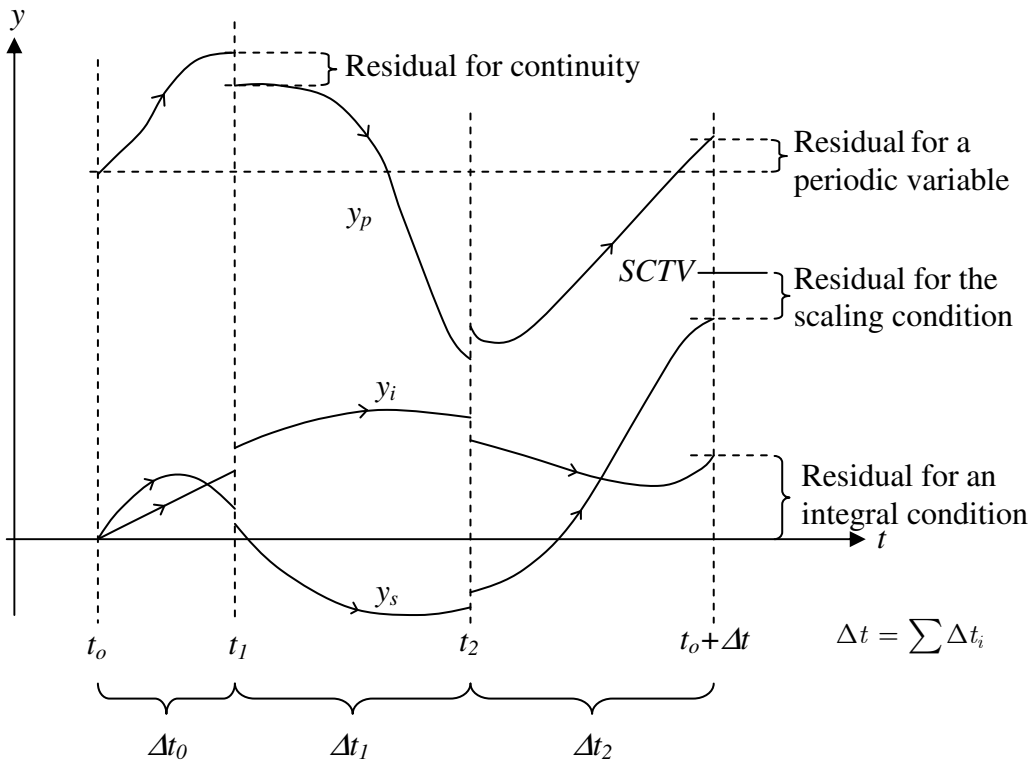


Fig. 39. Illustration of the multiple shooting method.

One advantage of the multiple shooting method over the single shooting method can be increased reliability for problems where integrations can easily fail. In such cases the subdivision of the cycle can help because the integration path through the cycle becomes less sensitive to individual initial values.

The splitting of the cycle into sub-intervals can also have advantages if the shooting method is parallelised.

For the models considered here it is so that almost all of the computational effort needed to perform single or multiple shooting is spent in the IVP solver which is used for integrating the model. Each residual update requires the entire cycle to be integrated once and each Jacobian update requires a larger number of integrations; the exact

number depends on the assumptions which can be made about the equations. The simulations needed for a Jacobian update are independent of each other and can be performed in parallel for both single and multiple shooting. For multiple shooting the integrations of the sub-intervals during a residual update can also be done in parallel.

The independent integrations each represent a relatively significant amount of computational work. This means that the shooting methods can be parallelised in a coarse grained way suitable for distributed memory platforms (clusters) by distributing the independent integrations between multiple processors.

For shooting methods discontinuous derivatives can be dealt with as events by the IVP solver used for integrating the equation system.

Shooting methods, just as finite difference methods, become cumbersome for finding periodic steady state solutions when fine discretisations are needed. For the shooting methods it turns out to be the cost of updating the Jacobian matrix which becomes dominant when the number of dynamic variables in a model increases. For both the single and multiple shooting methods the number of cycles which must be integrated in order to update a Jacobian is roughly proportional to the number of dynamic variables. The cost of performing each of the integrations, of course, also increases with the number of variables.

One positive aspect, however, is that the potential for parallelisation of Jacobian updates also increases with the number of dynamic variables. Another positive aspect is that the memory requirement is significantly smaller than for the finite difference method. When single shooting was performed for the 4^{th} refinement from Table 1 of the spatial discretisation for the Stirling machine model the LU-decomposed Jacobian for the shooting method required only 37 MB of memory. For the second refinement of the spatial discretisation the Jacobian required only 2.6 MB of memory which is insignificant on a present day computer. Even though additional memory was required by the IVP solver and the OESI the memory requirements were still very manageable.

4.2.4. Choice of method

Convergence acceleration or extrapolation methods appear to be the methods which could be most attractive when the number of dynamic variables in models are very large. Unfortunately, no cheap and stable way to solve integral conditions for constant wall temperatures has been found for these methods and therefore the methods do not appear suitable for the models at hand.

For the sizes of equation systems, which have been typical for the models used in this work, both finite difference methods and shooting methods appear to be attractive. A common feature of these methods is that they completely eliminate the slow asymptotic evolution of an IVP and instead offer the rate of convergence for a solution process for a non-linear equation system.

Previous work by Commiso (1994) showed that shooting methods can be competitive for finding periodic steady state solutions to models of Stirling machines. It is also attractive properties of shooting methods that discontinuous derivatives are uncomplicated to handle, and that the shooting methods can be parallelised with a coarse granularity suitable for execution on distributed memory platforms.

In this work it was chosen to focus on shooting methods. Significant efforts have been dedicated to parallelise and refine shooting methods. Finite difference methods have not been studied further.

4.3. Single and multiple shooting methods

4.3.1. Formulation for the single shooting method

To define a single shooting method we reformulate the conditions (17) for periodic steady state for $\underline{y_{p,0}}$ and $\underline{c_i}$ as a slightly modified system of residual equations. By introducing a scaling factor, λ, as an additional variable we can write the modified system of residual equations as:

$$\begin{aligned} \underline{res_p}\left(\underline{y_{p,0}}, \underline{c_i}\right) &= \begin{cases} \underline{y^s_{p,0}} - \underline{y^s_p}\left(t_0 + \Delta t; \underline{y_{p,0}}, \underline{c_i}\right) + (1-\lambda)\cdot \underline{y^s_{p,0}} \\ \underline{y^{ns}_{p,0}} - \underline{y^{ns}_p}\left(t_0 + \Delta t; \underline{y_{p,0}}, \underline{c_i}\right) \end{cases} \\ \underline{res_i}\left(\underline{y_{p,0}}, \underline{c_i}\right) &= \underline{0} - \underline{y_i}\left(t_0 + \Delta t; \underline{y_{p,0}}, \underline{c_i}\right) \\ res_s\left(\underline{y_{p,0}}, \underline{c_i}\right) &= SCTV - \lambda \cdot y_s\left(t_0 + \Delta t; \underline{y_{p,0}}, \underline{c_i}\right) \end{aligned} \tag{22}$$

The last term in the residual equations for $\underline{y^s_{p,0}}$ is a penalty factor that forces λ towards a value of one. We choose to solve this system of $N_p + N_i + 1$ equations for the $N_p + N_i$ values in $\underline{y_{p,0}}$ and $\underline{c_i}$ using a modified Newton-Raphson method.

We denote the Jacobian matrix for the equation system (22) as $\underline{\underline{J_s}}$ and the residuals of (22) as $\underline{r_s}$. Due to the complexity of the models that must be handled by the shooting method $\underline{\underline{J_s}}$ cannot be calculated analytically and must be calculated by numerical differencing. This means that we must perform, at least, one integration of the model in order to evaluate a column in $\underline{\underline{J_s}}$. When we compute $\underline{\underline{J_s}}$ we set $\lambda = 1$ as it will be for the solutions we seek.

In the iterations of the modified Newton-Raphson method we set $\lambda = 1$ and then solve a linear system and correct our guess for the solution as:

$$\underline{\underline{J_s}}.\begin{bmatrix} \underline{\delta y_{p,0}} \\ \underline{\delta c_i} \\ \delta\lambda \end{bmatrix} = -\underline{r_s} \quad , \quad \begin{bmatrix} \underline{y^s_{p,0}} \\ \underline{y^{ns}_{p,0}} \\ \underline{c_i} \end{bmatrix} = \begin{bmatrix} \underline{y^s_{p,0}} \\ \underline{y^{ns}_{p,0}} \\ \underline{c_i} \end{bmatrix} + \begin{bmatrix} \underline{\delta y^s_{p,0}} \\ \underline{\delta y^{ns}_{p,0}} \\ \underline{\delta c_i} \end{bmatrix} \tag{23}$$

The modification of the Newton-Raphson method is, hence, that we keep setting $\lambda = 1$ and thereby ignore $\Delta\lambda$.

4.3.2. Formulation for the multiple shooting method

In the single shooting method defined above we shoot across an entire cycle. As mentioned above we can extend the single shooting method to a multiple shooting method by slicing the cycle into multiple sub-intervals as shown in Fig. 39. Additional equations for the continuity of the solution are then needed.

To define a multiple shooting method we introduce cuts at N_{ip} internal points. If we choose to number the resulting $N_{ip}+1$ intervals from 0 to N_{ip} and label the variables belonging to interval j with subscript (j), then we can write the complete system of residual equations for the multiple shooting method as:

$$
\begin{aligned}
&\underline{res_p}\left(\underline{y_{(N_{ip}),p,0}}, \underline{y_{(0),p,0}}, \underline{c_i}\right) = \\
&\quad \begin{cases} \underline{y^s_{(0),p,0}} - \underline{y^s_{(N_{ip}),p}}\left(t_{N_{ip}} + \Delta t_{N_{ip}}; \underline{y_{(N_{ip}),p,0}}, \underline{c_i}\right) + (1-\lambda)\cdot \underline{y^s_{(0),p,0}} \\ \underline{y^{ns}_{(0),p,0}} - \underline{y^{ns}_{(N_{ip}),p}}\left(t_{N_{ip}} + \Delta t_{N_{ip}}; \underline{y_{(N_{ip}),p,0}}, \underline{c_i}\right) \end{cases} \\
&\underline{res_{(j)}}\left(\underline{y_{(j-1),0}}, \underline{y_{(j),0}}, \underline{c_i}\right) = \\
&\quad \begin{cases} \underline{y^s_{(j),p,0}} - \underline{y^s_{(j-1),p}}\left(t_{j-1} + \Delta t_{j-1}; \underline{y_{(j-1),p,0}}, \underline{c_i}\right) + (1-\lambda)\cdot \underline{y^s_{(j-1),p,0}} \\ \underline{y^{ns}_{(j),p,0}} - \underline{y^{ns}_{(j-1),p}}\left(t_{j-1} + \Delta t_{j-1}; \underline{y_{(j-1),p,0}}, \underline{c_i}\right) \\ \underline{y_{(j),i,0}} - \underline{y_{(j-1),i}}\left(t_{j-1} + \Delta t_{j-1}; \underline{y_{(j-1),0}}, \underline{c_i}\right) \\ y_{(j),s,0} - y_{(j-1),s}\left(t_{j-1} + \Delta t_{j-1}; \underline{y_{(j-1),0}}, \underline{c_i}\right) \end{cases} \qquad j = 1..N_{ip} \\
&\underline{res_{ic}}\left(\underline{y_{(N_{ip}),p,0}}, \underline{y_{(N_{ip}),i,0}}, \underline{c_i}\right) = \underline{0} - \underline{y_{(N_{ip}),i}}\left(t_{N_{ip}} + \Delta t_{N_{ip}}; \underline{y_{(N_{ip}),p,0}}, \underline{y_{(N_{ip}),i,0}}, \underline{c_i}\right) \\
&res_s\left(\underline{y_{(N_{ip}),p,0}}, \underline{y_{(N_{ip}),s,0}}, \underline{c_i}\right) = SCTV - \lambda \cdot y_{(N_{ip}),s}\left(t_{N_{ip}} + \Delta t_{N_{ip}}; \underline{y_{(N_{ip}),p,0}}, \underline{y_{(N_{ip}),s,0}}, \underline{c_i}\right)
\end{aligned}
\tag{24}
$$

We denote the Jacobian matrix for the equation system (24) as $\underline{\underline{J_{ms}}}$ and the residuals of (24) as $\underline{r_{ms}}$. In the multiple shooting method we also choose to solve our residual equation system of $N_{ip}\cdot(N_p + N_i + 1)$ equations for the $N_{ip}\cdot(N_p + N_i + 1) - 1$ values in $\underline{y_{(0),p,0}}$, $\underline{c_i}$, and $\underline{y_{(j),0}}$, $j = 1..N_{ip}$ using a modified Newton-Raphson method. In the modified Newton-Raphson iterations of the multiple shooting method we also keep setting $\lambda = 1$ and we solve a linear system and correct our guess for the solution as:

$$
\begin{aligned}
&\underline{\underline{J_{ms}}} \cdot \begin{bmatrix} \underline{\delta y_{(0),p,0}} \\ \underline{\delta y_{(j)}}, j = 1..N_{ip} \\ \underline{\delta c_i} \\ \delta\lambda \end{bmatrix} = -\underline{r_{ms}} \quad , \\
&\begin{bmatrix} \underline{y^s_{(0),p,0}} \\ \underline{y^{ns}_{(0),p,0}} \\ \underline{y_{(j),0}} \\ \underline{c_i} \end{bmatrix} = \begin{bmatrix} \underline{y^s_{(0),p,0}} \\ \underline{y^{ns}_{(0),p,0}} \\ \underline{y_{(j),0}} \\ \underline{c_i} \end{bmatrix} + \begin{bmatrix} \underline{\delta y^s_{(0),p,0}} \\ \underline{\delta y^{ns}_{(0),p,0}} \\ \underline{\delta y_{(j),0}} \\ \underline{\delta c_i} \end{bmatrix} \quad \} j = 1..N_{ip}
\end{aligned}
\tag{25}
$$

When the equations are ordered as in (24) then the sparse structure of $\underline{\underline{J_{ms}}}$ takes the form (26) where $\underline{\underline{I}}$ is the identity matrix and the X symbols mark other non-zero entries:

$$\underline{\underline{J_{ms}}} = \begin{array}{c} \begin{matrix} \underline{y_{(0),p,0}} & \underline{y_{(1),0}} & \underline{y_{(2),0}} & \underline{y_{(3),0}} & \cdots & \underline{y_{(N_{ip})}} & \underline{c_i} & \lambda \end{matrix} \\ \left[\begin{matrix} \underline{\underline{I}} & & & & & \underline{\underline{X}} & \underline{\underline{X}} & \underline{X} \\ \underline{\underline{X}} & \underline{\underline{I}} & & & & & \underline{\underline{X}} & \underline{X} \\ & \underline{\underline{X}} & \underline{\underline{I}} & & & & \underline{\underline{X}} & \underline{X} \\ & & \underline{\underline{X}} & \underline{\underline{I}} & & & \underline{\underline{X}} & \underline{X} \\ & & & \ddots & \ddots & & \vdots & \vdots \\ & & & & \underline{\underline{X}} & \underline{\underline{I}} & \underline{\underline{X}} & \underline{X} \\ & & & & & \underline{\underline{X}} & \underline{\underline{X}} & \underline{X} \\ & & & & & \underline{X} & X & X \end{matrix}\right] \end{array} \begin{matrix} \underline{res_p} \\ \underline{res_{(1)}} \\ \underline{res_{(2)}} \\ \underline{res_{(3)}} \\ \vdots \\ \underline{res_{(N_{ip})}} \\ \underline{res_i} \\ res_s \end{matrix} \tag{26}$$

4.3.3. Reducing the dimension of the boundary value problem with single shooting

For the single shooting method we can sometimes use insight into the different time scales of the physical phenomena being modelled to reduce the dimension of the BVP handled by the shooting method.

For some models it is so, that when a simulation is started as an initial value problem then the velocity and pressure distributions will very quickly reach a sort of equilibrium with the remainder of the solution.

An example of this can be observed when a simulation of a Stirling machine is performed as an initial value problem. In such a simulation the pressure and velocity distributions may undergo fast initial transients but they will very quickly decay onto distributions which are given more or less by the movement of the pistons and the flow friction in the heat exchangers and the regenerator. The initial fast transients will typically last only a small fraction of the cycle period. Other variables, such as the regenerator matrix temperatures, will evolve much more slowly.

When such a difference in the time scales of the periodic variables exists it is possible to exclude the fastest components of the solution from the single shooting method. The idea is then to apply the shooting method only to the integral conditions and the slowly evolving components of $\underline{y_p}$, such as the regenerator matrix temperatures and the gas temperatures, and to perform fixed point iteration on the excluded rapidly evolving components of $\underline{y_p}$.

If we choose to exclude the variables which behave like pressures and velocities from the single shooting method then we must also remove the equations corresponding to the excluded variables and y_s from (22). We then perform fixed point iteration on the excluded variables by changing the corrections for $\underline{y_p}$ in the modified Newton-Raphson iterations in (23) as follows:

$$\begin{bmatrix} \underline{y^s_{p,0}} \\ \underline{y^{ns}_{p,0}} \\ \underline{y^{ns}_{p,0}} \\ \underline{c_i} \end{bmatrix} = \begin{bmatrix} \underline{y^s_{p,0}} \\ \underline{y^{ns}_{p,0}} \\ \underline{y^{ns}_{p,0}} \\ \underline{c_i} \end{bmatrix} + \begin{bmatrix} \dfrac{SCTV}{y_s\left(t_0+\Delta t;\underline{y_{p,0}},\underline{c_i}\right)} \cdot \underline{y^s_p}\left(t_0+\Delta t;\underline{y_{p,0}},\underline{c_i}\right) - \underline{y^s_{p,0}} \\ \left(\underline{y^{ns}_p}\left(t_0+\Delta t;\underline{y_{p,0}},\underline{c_i}\right) - \underline{y^{ns}_{p,0}}\right) \\ \dfrac{\delta \underline{y^{ns}_{p,0}}}{\delta \underline{c_i}} \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix}a\\b\end{matrix}}\right\} \text{excluded from shooting} \\ \left.\vphantom{\begin{matrix}a\\b\end{matrix}}\right\} \text{included in shooting} \end{matrix} \tag{27}$$

By reducing the number of variables included in the shooting method the effort needed to compute $\underline{\underline{J_s}}$ is reduced.

It has been observed that the convergence of shooting methods suffers significantly if both the pressures and velocities are excluded when finding periodic steady state solutions to models of pulse tube coolers. This could be because pressure transients take somewhat longer to equalise across the inertance tube. For models of pulse tube coolers it appears to be optimal to exclude only the velocities from the shooting.

Unfortunately, there appears to be no obvious way to extend the method described here for reducing the dimension of the BVP to the multiple shooting method. The problem is that there appears to be no efficient way to perform fixed point iteration on the excluded variables simultaneously in multiple intervals.

4.3.4. Comparison of work for shooting methods

The shooting methods described above require the entire cycle of length Δt to be integrated in order to update the residuals. But the total amounts of running time of the simulated machine that must be integrated in order to compute a Jacobian matrix differ for the methods.

If we calculate the elements of the Jacobian using one sided differences and if we already know the current residuals, then we need to integrate the entire cycle $N_p + N_i + 1$ times in order to compute $\underline{\underline{J_s}}$ for the single shooting method. If we reduce the dimension of the boundary value problem by omitting the variables which behave like pressures and velocities then we reduce the required number of integrations by the number of pressures and velocities plus one; the plus one comes from λ which is not used in the shooting method when the pressures are excluded.

Determining the number of integrations required to calculate $\underline{\underline{J_{ms}}}$ for the multiple shooting method is slightly more complex.

Calculating the non-zero elements corresponding to the X symbols in (26) requires $(N_{ip}+1)\cdot(N_p+N_i+1)+N_{ip}\cdot(N_i+1)$ integrations of sub-intervals. The term $(N_{ip}+1)\cdot(N_p+N_i+1)$ is the effort needed to integrate the entire cycle N_p+N_i+1 times, and it corresponds to the effort needed to compute $\underline{\underline{J_s}}$. The term $N_{ip}\cdot(N_i+1)$ comes from the continuity equations for the integral conditions and the scaling condition at the internal points, and it represents an overhead relative to the effort needed to compute $\underline{\underline{J_s}}$.

For some models the overhead corresponding to the term $N_{ip}\cdot(N_i+1)$ can be fully or partially avoided. If the values of $\underline{y_i}$ and y_s do not affect $\underline{y_p}$ during an integration then $N_{ip}\cdot(N_i+1)$ perturbations of $\underline{y_i}$ and y_s at the internal points in the cycle can be omitted during updates of $\underline{\underline{J_{ms}}}$. Any changes in the initial values of $\underline{y_i}$ and y_s in a sub-interval will then transfer directly to the final values of $\underline{y_i}$ and y_s in the same sub-interval. The corresponding elements of unit magnitude can be inserted directly in $\underline{\underline{J_{ms}}}$.

Here it has been chosen to skip the perturbations of $\underline{y_i}$ and y_s only at the first $N_{ip}-1$ internal points so that the effects of structural heat conduction calculations, which change the values of $\underline{y_i}$ after integration of the last sub-interval (interval number N_{ip}), can be included in the Jacobian matrix.

Even when the total machine running time that must be integrated is approximately the same for Jacobian updates in the single and multiple shooting methods, $\underline{\underline{J_{ms}}}$ can still be more costly to evaluate than $\underline{\underline{J_s}}$ because of the increased number of simulation start ups.

At the beginning of each integration there can be a fast transient in the pressure and velocity distributions. During such a transient the IVP solver will take very small steps. The transient will hence increase the needed number of steps pr. integration and hence the computational effort. In practice, the perturbing during a Jacobian update of variables which behave like pressures also induces fast transients.

There can also be a minor overhead due to restarting the IVP solver and its step size control algorithm.

4.3.5. Reducing computational work

Fast transients at the beginning of integrations will cause an overhead when $\underline{\underline{J_s}}$ or $\underline{\underline{J_{ms}}}$ are updated. This overhead can be especially significant when $\underline{\underline{J_s}}$ and, in particular, $\underline{\underline{J_{ms}}}$ are computed from a solution with poor initial guess for the pressure distribution so that every integration begins with a large fast transient.

Smoothing runs

The overhead from fast transients can be reduced by filtering the major fast transients from the current guess for the solution before updating a Jacobian. The filtering can be performed by simulating all sub-intervals in the cycle once and then cyclically shifting the final values to be new initial values in the following way:

$$\begin{aligned} &\underline{y_{(N_{ip}),p}}\left(t_{N_{ip}}+\Delta t_{N_{ip}}\right) \quad \text{transfers to} \quad \underline{y_{(0),p,0}} \\ &\underline{y_{(j-1)}}\left(t_{j-1}+\Delta t_{j-1}\right) \quad \text{transfers to} \quad \underline{y_{(j),0}} \quad ,j=1..N_{ip} \end{aligned} \tag{28}$$

Such simulations will be denoted as *smoothing runs* from hereon.

Smoothing runs which scale the solution

Smoothing runs can also be applied to filter fast transients from the initial guess for the solution before the shooting method is applied. In that context the smoothing runs can be extended so that they also help to scale $\underline{y_p^s}$ to the correct level before the shooting method is applied. This can be done by modifying (28) to the form (29):

$$\left.\begin{array}{lll}
\dfrac{SCTV}{y_s\left(t_{N_{ip}}+\Delta t_{N_{ip}}\right)}\cdot \underline{y^s_{(N_{ip}),p}}\left(t_{N_{ip}}+\Delta t_{N_{ip}}\right) & \text{transfers to} & \underline{y^s_{(0),p,0}} \\
\underline{y^{ns}_{(N_{ip}),p}}\left(t_{N_{ip}}+\Delta t_{N_{ip}}\right) & \text{transfers to} & \underline{y^{ns}_{(0),p,0}} \\
\dfrac{SCTV}{y_s\left(t_{N_{ip}}+\Delta t_{N_{ip}}\right)}\cdot \underline{y^s_{(j-1),p}}\left(t_{j-1}+\Delta t_{j-1}\right) & \text{transfers to} & \underline{y^s_{(j),p,0}} \\
\underline{y^{ns}_{(j-1),p}}\left(t_{j-1}+\Delta t_{j-1}\right) & \text{transfers to} & \underline{y^{ns}_{(j),p,0}} \\
\underline{y_{(j-1),i}}\left(t_{j-1}+\Delta t_{j-1}\right) & \text{transfers to} & \underline{y_{(j),i,0}} \\
y_{(j-1),s}\left(t_{j-1}+\Delta t_{j-1}\right) & \text{transfers to} & y_{(j),s,0}
\end{array}\right\} j=1..N_{ip} \qquad (29)$$

Recycling of shooting Jacobians

When iterating the shooting method it is usually possible to recycle the Jacobian matrix for multiple iterations, and sometimes even between multiple solutions to the same model, if convergence remains satisfactory.

A robust criterion for the recycling of Jacobian matrices is needed. Here it has been chosen to base the criterion on the L^2-norms of four groups of residuals: $\underline{res_p}$, $\underline{res_i}$, res_s, and $\underline{res_{(j)}},\ j=1..N_{ip}$. When none of the four L^2-norms have decreased better than a threshold value in either of the last two iterations then the Jacobian matrix is updated. This requirement is not very strict and it leaves the possibility that the solution can get stuck or diverge at some point during the iterations with the individual residuals norms alternately increasing and decreasing in such a way that the Jacobian matrix is not updated. To rule out this possibility the maximum residual is continuously monitored and if it has not decreased in 10 iterations then the Jacobian is updated.

When the Jacobian matrix for the shooting method must be updated the current guess for the solution is rolled back to the solution which produced the smallest maximum residual since the last Jacobian update. This solution is then filtered with a smoothing run before computing the Jacobian.

4.3.6. Parallelisation of shooting methods

When the shooting methods discussed above were applied to models of Stirling machines and pulse tube coolers nearly all the CPU-time needed by the methods was spent in the IVP method used for integrating the model. Hence the efforts to parallelise the shooting methods were concentrated on the parts of the methods that require simulations to be performed: Performing smoothing runs, calculating residuals, and updating Jacobians.

For both the single and multiple shooting methods one integration of all sub-intervals in the cycle is needed to update the residuals or to perform a smoothing run as shown in (28) and (29). Hence the shooting methods use of the IVP solver was encapsulated in two subroutines: A subroutine that computes $\underline{y_{(j)}}\left(t_j+\Delta t_j\right),\ j=0..N_{ip}$ and a subroutine that computes $\underline{\underline{J_s}}$ or $\underline{\underline{J_{ms}}}$. Parallel and sequential versions were made of these two subroutines.

The two subroutines that perform integrations for the shooting methods can be thought of as being parallelised with two different levels of granularity.

At the fine grained level the IVP method was parallelised to use multi-threading as discussed in section 3.3.6. The implemented multi-threading can only be used on a shared memory platform with an OpenMP compliant compiler. As demonstrated in 3.6.2 the maximum speedup which can be achieved by multi-threading is significant but somewhat limited for the relevant size of equation systems.

At a much coarser level of granularity the two subroutines have been parallelised by making them distribute independent integrations of sub-intervals between multiple processes. The communication between the processes has been implemented using the Message Passing Interface, MPI.

In the remainder of this chapter we consider only the coarse level parallelisation.

Throughout this chapter a clear distinction is made between *jobs* and *tasks*:

- A *job* corresponds to one solution to the BVP.
- A *task* corresponds to the integration of one sub-interval for the shooting method.

Example: The batch job, which was used to validate the Stirling machine model against experimental data for the SM5 engine in section 3.9.1, contained 28 *jobs* and each individual job required a number of *tasks* to be performed in order to find the solution using a shooting method.

For the parallelised subroutine, which integrates all sub-intervals in the cycle, the number of tasks which can be distributed between different processes is equal to $N_{ip}+1$. This subroutine can hence only achieve a parallel speedup if $N_{ip}>0$.

For the subroutine which updates the Jacobian the number of tasks which can be distributed increases as a function of both N_{ip} and of the number of variables in the model. The exact number of tasks which can be distributed also depends on the assumptions which can be made about the equation system as discussed in section 4.3.4. In general the subroutine which updates Jacobians for the shooting methods will always have a number of tasks which can be performed in parallel.

For both of the subroutines the number of tasks which can be distributed increases with N_{ip} and therefore the achievable speedup from parallelisation should also increase with N_{ip}.

The parallelisation of the two subroutines was done using the pool of tasks paradigm (Wilkinson and Allen, 1999). With this paradigm one master process tries to keep a number of worker processes busy by distributing independent tasks to idle processes until all tasks in the pool of tasks is complete.

The pool of tasks approach is advantageous because it allows the worker processes to work independently and asynchronously. This means the parallel program can be relatively insensitive to inhomogeneities in the parallel computing environment. Such inhomogeneities can be processors running at different speeds, communication between processors running at different speeds, or it could be some processors being slightly loaded with other work. The main disadvantage of the pool of tasks approach, when it is possible to apply it, is that all communication goes through the master process. If the communication is excessive the master can become a bottleneck.

In the implementation, which is tested below, it was the process running the shooting algorithm that acted as the master process. The independent tasks were the integrations of (sub-intervals of) the cycle. The worker processes performed no other services than integrations.

4.4. Overview of tests

The shooting methods described above were first implemented and tested early in this work. At that time the modelling formulation used in this work for modelling Stirling machines was significantly less refined than the approach described in chapter 3. The most important difference in this context turns out to be that the momentum balance had been simplified by neglecting the inertia of the gas. When the modelling approach was refined to include the inertia of the gas in the momentum equation the performance of the multiple shooting methods changed significantly.

The emphasis in the tests presented here are on the performance of the shooting methods for a model like the Stirling machine model described in section 3.4, i.e. the performance for models which include the full momentum balance. But tests, where the shooting methods were applied to a model with a simplified momentum balance, have been included in order to illustrate the impact on the performance of the shooting methods from including the inertia of the gas.

The tests performed on the single and multiple shooting methods can be categorised in the following way:

1. **Tests on a model with a simplified momentum balance.** The purposes of these tests were to illustrate: 1) the performance of the shooting methods for a model where the momentum balance has been simplified by neglecting the inertia of the gas, and 2) the performance of the parallel methods on a cluster with a slow interconnect between the nodes. The computational overhead from using multiple shooting instead of single shooting was studied. The scalabilities of the parallel sections and of the different shooting methods were studied.
2. **Tests on a model with a complete momentum balance.** The purpose of these tests was to illustrate the performance of the shooting methods for a Stirling machine model very similar to the model described in section 3.4. The shooting methods and the integration to convergence method were profiled in sequential runs. The scalabilities of the shooting methods up to 33 processors were measured on a shared memory computer.

The parallel scalabilities of the tested methods were assessed in terms of the absolute speedups that were achieved when the methods used multiple processors. The absolute speedup was defined as the sequential running time divided by the parallel running time.

The parallel and sequential methods were compared on basis of the relative speedups. The relative speedup achieved in a given experiment was defined as the sequential running time of the fastest sequential method divided by the running time in the given experiment.

The individual tests and the results are described in the sections that follow.

4.5. Tests on a model with a simplified momentum equation

The tests described here are intended to illustrate for a model with a simplified momentum balance:

- The overhead from splitting the cycle into multiple intervals when integrating the entire cycle on and when updating a Jacobian matrix for the shooting method.
- The scalabilities of the parallelised subroutines of the shooting methods on a distributed memory platform with a 100 Mbit/s Ethernet interconnect.
- The relative scalabilities of the single and multiple shooting methods on a distributed memory platform.

4.5.1. Method

The model and the spatial discretisation
A model of the SM5 engine was used for the tests. The model was based on the methods described in the accompanying *Paper A*. The model was significantly different from the Stirling machine model described in section 3.4. The primary differences from the model described in section 3.4 were:

- The inertia of the gas was neglected. The instantaneous mass flow rates between the control volumes in the spatial discretisation were calculated from the instantaneous pressure differences between the control volumes. Because of this there were no ODEs for the momentum balance.
- The appendix gap was not included in the model.
- The inner wall temperatures in some of the heat exchangers were prescribed explicitly so that there were no integral conditions for the inner wall temperatures. The walls of the regenerator canister were not accounted for; only the regenerator matrix was modelled. These simplifications led to fewer integral conditions.
- The model used masses and energies directly as integration variables for the governing equations.

In the tests the spatial discretisation in the model contained 61 control volumes. This discretisation resulted in $N_p = 151$ and $N_i = 16$ corresponding to a total of 168 variables in the IVP.

The platform
For these tests a computer cafe at the Technical University of Denmark was used as a non-dedicated cluster. The computer cafe contained 14 Dell Optiplex PCs equipped with 1.4 GHz Intel Pentium 4 processors with 512 KB Level 2 cache and 512 MB RAM. The PCs were connected via 100 Mbit/s Ethernet and were running the Microsoft Windows 2000 Professional operating system. In order to use the computer room as a cluster the NT-MPICH (LfBS RWTH Aachen, 2002) open source implementation of MPI was installed after being slightly modified to accommodate the security requirements of the computer room.

The test jobs
Two different test runs were used:

- **Jacobian update:** These test runs were started from periodic steady state solutions. One residual update was performed, the Jacobian was then updated and LU-decomposed, and a single iteration was performed.

- **7 MPa to 8 MPa:** These experiments were started from a periodic steady state solution with 7 MPa mean pressure, and with a precalculated Jacobian matrix. The precalculated Jacobian matrix was used for finding a new solution at 8 MPa mean pressure. The Jacobian was not updated in these tests and the only integrations performed were integrations of the entire cycle.

The tests
First both test jobs were completed sequentially for the single shooting method and for the multiple shooting method with the cycle divided into 2, 5, 10, 20, 50, 100, 200, and 500 sub-intervals. No variables were excluded from the shooting.

From these experiments the times required to update the Jacobian matrix and the average time to simulate one cycle were recorded. The number of iterations needed for the test job *7 MPa to 8 MPa* were also recorded. The timings recorded in the experiments were divided by the timings recorded for the single shooting method so that, for instance, a ratio of 1.1 meant that there was a 10 % overhead relative to the single shooting method.

The test jobs were then completed in parallel for the single shooting method and for the multiple shooting method with 100 sub-intervals in the cycle. 2 to 14 processors were used in the parallel tests. On the PCs it was possible to run multiple MPI processes with different priorities on the same processor. This feature was used to run both the master process and a worker process with a lower priority on the same processor. In this way the processor running the master process could be fully utilised.

The numbers of tasks in a simulation of a cycle and in a Jacobian update are summarised in Table 9 for the single shooting method and the multiple shooting method with 100 sub-intervals in the cycle.

Name of method	Variables excluded from shooting	N_{ip}	Tasks in simulation of a cycle	Tasks in a Jacobian update
Single shooting	None	0	1	168
Multiple shooting, 100 int.	None	99	100	16817

Table 9. Parallel characteristics of methods included in the numerical tests on a model with a simplified momentum balance.

From these experiments the total running time and the time spent in the parallel sections was recorded. The absolute speedups for the parallel sections and the relative speedups for the methods were computed from the recorded timings.

Solver settings
The semi implicit GERK scheme of order three by Thomsen (2002) was used for integrating the model. A relative tolerance of 10^{-7} was used for the error pr. step taken by the GERK method, and a relative residual tolerance of 10^{-11} was used by the Newton-Raphson solver at the stages of the GERK method.

4.5.2. Numerical results and discussion

Fig. 40 shows the measured overhead for simulations of an entire cycle and for updates of the Jacobian matrix for the shooting method with 1 to 500 sub-intervals in the cycle. The figure shows that the overhead for both integration of a cycle and for Jacobian

updates appeared to be approximately proportional to the number of sub-intervals in the cycle. The figure also shows that the overhead was insignificant for less than 20 sub-intervals.

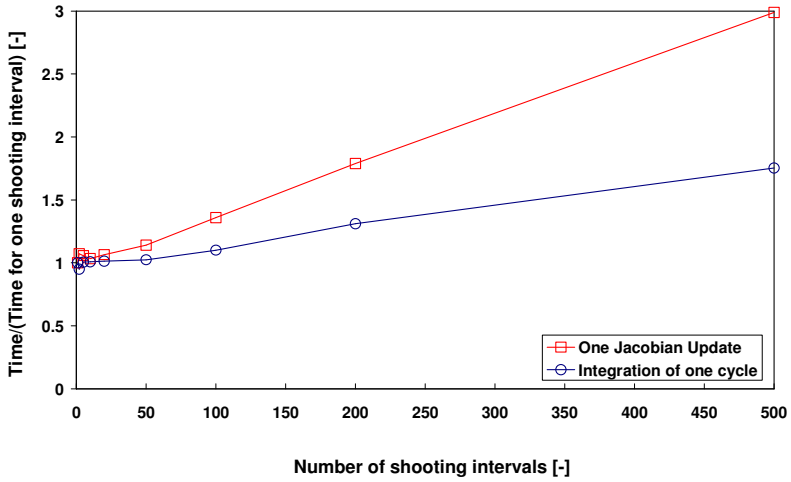


Fig. 40. Overhead from subdivisions of the cycle for a model with a simplified momentum equation.

Table 10 shows that the number of iterations needed in the *7 MPa to 8 MPa* test job varied between three and five for all the tested shooting methods. Table 10 does not appear to show any strong and consistent dependence of the speed of convergence on the number of sub-intervals in the cycle.

Number of sub intervals	1	2	5	10	20	50	100	200	500
Shooting iterations 7 to 8 MPa	3	4	3	3	5	3	4	4	4

Table 10: Number of iterations needed by the shooting methods for the test job *7 MPa to 8 MPa*.

Fig. 41 shows the absolute speedups achieved for the parallel sections for the single shooting method and for multiple shooting with 100 sub-intervals in the cycle. The figure shows that the parallel sections scaled very well with the number of processors.

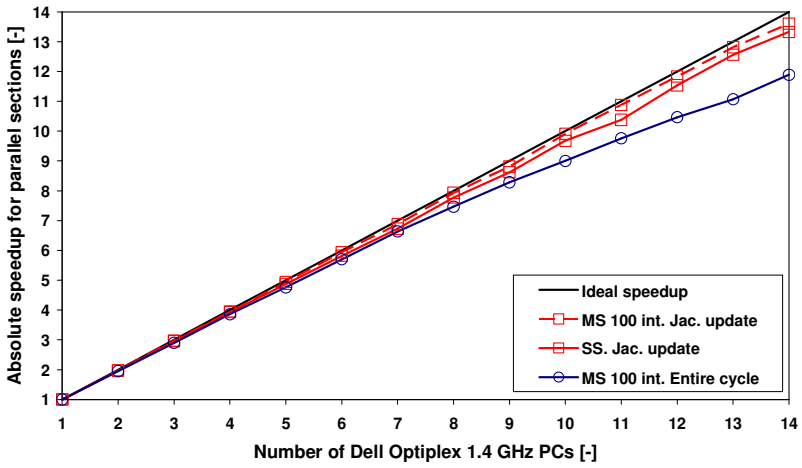


Fig. 41. Absolute speedups for parallel sections for the single shooting method and the multiple shooting method with 100 intervals in the cycle.

By correlating with Table 9 it can be seen that the scalabilities of the parallel sections increased with the number of tasks in each parallel section. This indicates that, as

expected, the load balancing between the processors improved with the number of tasks that could be distributed.

The almost perfect scaling of the Jacobian update for the multiple shooting method with 100 sub-intervals in the cycle showed that the communications overhead was minimal during the test.

The integrations of sub-intervals of the cycle required 0.5 seconds of CPU time on average. When using 14 processors this meant that every second the master process had to receive approximately 28 messages with results and send approximately 28 messages with new jobs. The messages contained a number of floating point values close to the number of dynamic variables in the model and a few additional flags and settings; the largest messages were less than 2 kB in size. 56 messages of 2 kB size every second corresponds to a bandwidth need of only approximately 0.9 MBit/s for the master process. The results show that the 100 Mbit/s interconnect was perfectly adequate for handling such a low communications load. The low communications load was due to the coarse grained parallelisation of the shooting methods.

Fig. 42 shows a comparison of the relative speedups achieved for the single shooting method and the multiple shooting method with 100 sub-intervals in the cycle for the two test jobs described in section 4.5. Since the single shooting method was the fastest sequential method for both test jobs the results for single shooting have been used as numerator when computing the relative speedups.

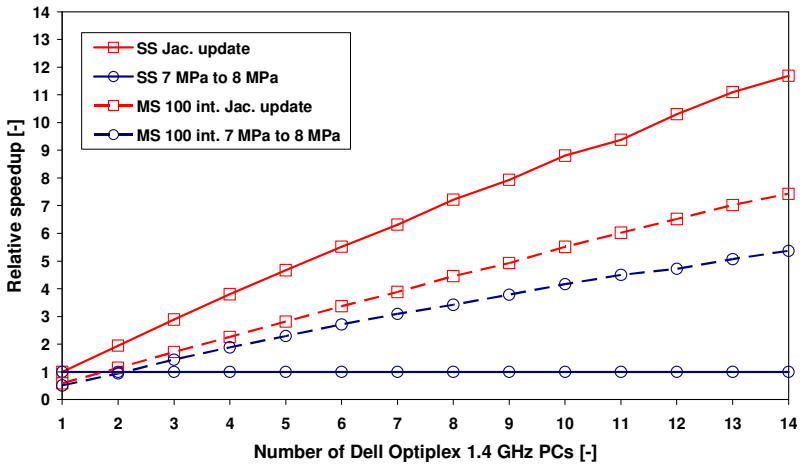


Fig. 42. Comparison of single shooting and multiple shooting with 100 intervals for a model with a simplified momentum balance for the two test jobs described in section 4.5.1.

Fig. 42 shows that the optimal choice of method depended both on the test problem and on the number of processors. For the test job, which consisted mainly of a Jacobian update, the single shooting method was always preferable. For the test job, which consisted of iterations, the multiple shooting method with 100 intervals in the cycle was fastest when 3 or more processors were used.

4.5.3. Conclusions

In the tests where the shooting methods were applied to the model with a simplified momentum balance it was found that:

- The division of the cycle into sub-intervals caused an overhead when the entire cycle was integrated and when the Jacobian matrix was updated. The overhead was insignificant if 20 or less intervals were used. With 100 sub-intervals the overhead varied between 10 and 35 %.
- There was no strong and consistent dependence of the speed of convergence on the number of sub-intervals in the cycle.
- The parallelised subroutines scaled very well with the number of processors on a cluster where the PCs were connected using 100 Mbit/s Ethernet. The results indicated that the communications overhead was minimal. This indicates that the coarse grained parallelisation of the shooting methods is suitable for distributed memory platforms.
- The optimal choice of method depended on the balance between the time spent on integrating the entire cycle and the time spent on Jacobian updates in the problem to be solved. If the majority of the time was spent on Jacobian updates then the single shooting method was preferable because the method had minimal overhead and already scaled well to a moderate number of processors. If a significant fraction of the time was spent on iterations, i.e. on integrating the entire cycle, then the multiple shooting methods were competitive.

4.6. Tests on a model with a full momentum equation

The tests described here are intended to illustrate the performance and scalabilities of the shooting methods for finding periodic steady state solutions to a model with a complete momentum balance.

4.6.1. Method

The model and the spatial discretisation
The shooting methods were tested using a Stirling machine model very similar to the model described in section 3.4. The model was configured to model the SM5 Stirling engine. The model used for these tests included the full momentum balance for the gas and it contained the same components as the model described in section 3.4. The model described in section 3.4 does contain some improvement over the model used for these tests. But the characteristics of the model used for these tests with respect to the performance of the shooting methods does not differ significantly from the model described in section 3.4.

In these tests the spatial discretisation in the model contained 93 solid line control volumes. With this discretisation $N_p = 319$ and $N_i = 78$ giving a total of 398 variables in the IVP. The discretisation used in the tests hence had 5 % more variables in the IVP than the first refinement of the spatial discretisation in Table 1.

The test job
A batch job with the 28 simulations of the SM5 engine, which were also used for the experimental validation of the Stirling machine model in section 3.9.1, were used as test job.

The test job was considered difficult with regards to recycling of the Jacobians for the shooting methods because the solutions spanned wide ranges of temperatures,

pressures, and ratios between gas and matrix heat capacities, and because flow friction was much larger for nitrogen as working gas than for helium.

In the test job the measure for the accuracy of the solutions was based on the imbalance in the overall energy balance for the engine. For each solution shooting was terminated when the absolute value of the imbalance in the overall energy balance was less than 0.1 Watts.

The test job was started without a Jacobian from a common reference solution not included in the test job.

The computer platform

The tests were run on a Sun Fire 15000 server with 42 UltraSparc III processors operating at 1.05 GHz and running the Solaris 9 operating system. The Sun Fire 15000 is a shared memory machine and the communications overhead from MPI communications should thus be minimal. The timing results contain a small amount of noise because the tests were run in a multi-user environment.

The tests

To evaluate the performance of the methods the test job was completed in the following tests:

1. Integration to convergence without acceleration towards periodic steady state.
2. Sequential tests of the shooting methods.
3. Parallel tests of the shooting methods.

The first test was performed to have a base case against which to compare the shooting methods. The test was carried out by excluding all elements of $\underline{y_p}$ from the single shooting method and then iterating as described by (27). Thereby the shooting method was used to update the wall temperatures after each simulated cycle so that only one integration to convergence was needed for each of the 28 simulations in the batch job.

The measurements of the sequential performance of the shooting methods were carried out in order to compare the computational costs of the methods without including any overhead from communications and load balancing.

For the multiple shooting method it was chosen to place, respectively, 1, 3, 7, and 15 internal points in the cycle. To facilitate good load balancing the internal points were placed so that the resulting sub-intervals of the cycle required approximately the same computational effort to simulate. The placement of the internal points was made by integrating a periodic steady state cycle and storing the position in the cycle and the wall time at every step taken by the IVP solver. The positions of the internal points were then determined by interpolation in the stored data.

For both integration to convergence and for the shooting methods the average time required to integrate one complete cycle and the average time needed to perform a Jacobian update were computed from the measured data.

The parallel tests were performed to measure the scalabilities of the shooting methods on a parallel computer platform. In the tests the total running times, profiles, and the numbers of evaluations were recorded.

The scalabilities of the shooting methods were measured by running them on 3, 5, 9, 17, and 33 processors. This corresponds to 2, 4, 8, 16, and 32 worker processes being available to perform tasks for the master process. The scalabilities of the shooting

methods were assessed by the absolute speedups. The methods were compared using the relative speedups achieved by the methods.

For each of the shooting methods the solutions be bitwise identical independently of the number of worker processes. For each of the shooting methods the sequential and parallel runs therefore must have expended exactly the same amount of work in the integrations of sub-intervals.

The numbers of tasks in a simulation of a cycle and in a Jacobian update are summarised for the tested methods in Table 11.

Name of method	Variables excluded from shooting	N_{ip}	Tasks in simulation of a cycle	Tasks in a Jacobian update
Integration to convergence	$\underline{y_p}$ (319 vars.)	0	1	78
Single shooting, excl. $\underline{P}$, $\underline{V}$	$\underline{p}$, $\underline{\mathrm{V}}$ (191 vars.)	0	1	206
Single shooting	None	0	1	398
Multiple shooting, 2 int.	None	1	2	875
Multiple shooting, 4 int.	None	3	4	1671
Multiple shooting, 8 int.	None	7	8	3263
Multiple shooting, 16 int.	None	15	16	6447

Table 11. Parallel characteristics of methods included in the numerical tests on a model with a full momentum balance.

Solver settings
The semi implicit GERK scheme by Thomsen (2002) was used for integrating the model. A relative tolerance of 10^{-7} was used for the error pr. step taken by the GERK method, and a relative residual tolerance of 10^{-11} was used by the Newton-Raphson solver at the stages of the GERK method.

The threshold value for the decrease in the residuals, which was used for controlling the recycling of Jacobians for the shooting methods, was set to 0.85 for the single and multiple shooting methods. For integration to convergence the criterion was changed so that the Jacobian was updated only if the imbalance in the overall energy balance had not decreased in two consecutive iterations.

4.6.2. Numerical results and discussion

Integration to convergence
Fig. 43 shows an example of how the residual norms and the imbalance in the overall energy balance of the engine decreased during integration to convergence for the model with a full momentum balance. This particular solution had helium at 6.5 MPa mean pressure as working gas.

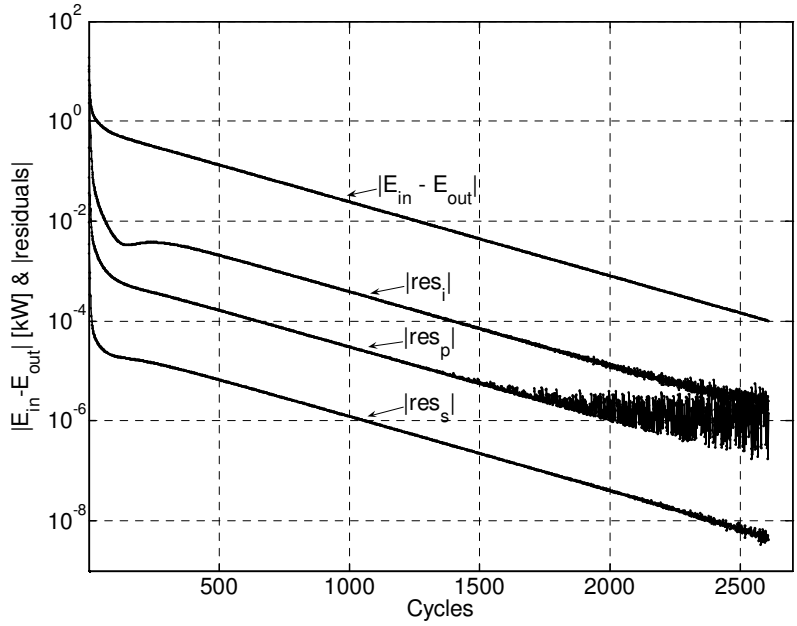


Fig. 43: Example of the decrease in the residual norms and the imbalance in the energy balance of the engine when performing integration to convergence.

Fig. 43 shows that simulation of approximately 2600 cycles was needed to achieve the desired tolerance. The SM5 engine operates at 1025 RPM so this corresponds to approximately 2.5 minutes running time for the engine. After the initial 500 cycles the residual norms and the imbalance in the overall energy balance of the engine appeared to decrease linearly in the semi-logarithmic plot in Fig. 43. This indicated that the solution approached periodic steady state exponentially.

After approximately 1500 cycles random noise began to show in the residual norms indicating that the changes from the evolution of the solution during each cycle approached the level of noise due to the chosen tolerance for the IVP method. The noise was most significant for $\underline{res_p}$ which contained the largest number of variables and included the fast variables for pressures and velocities. The imbalance in the overall energy balance for the engine, though, still decreased monotonously.

If the residual norms had been used to determine when $\underline{\underline{J_s}}$ should be updated, the observed noise would have caused $\underline{\underline{J_s}}$ to be updated more frequently than what was needed here to obtain the desired accuracy. Hence the choice of basing the criterion for recycling Jacobians on the imbalance in the overall energy balance was appropriate for this test.

The number of iterations required to reach the desired accuracy appeared to be strongly influenced by the ratio of the heat capacity of the gas to the heat capacity of the regenerator matrix. Integration to convergence with helium at 5 MPa and 8 MPa mean pressures required approximately 3200 and 2100 cycles, respectively. Integration to convergence with nitrogen at 6.5 MPa and 8 MPa mean pressure required approximately 1400 and 1200 cycles to converge.

Comparison of sequential methods

The results from the sequential tests of the shooting methods and of integration to convergence are summarised in Table 12.

	Running time		Profile					Evaluations				Avrg. cost of evals.	
Method	Time [h]	Index [-]	Resid. [%]	Sm. r. [%]	Jac. [%]	LA [%]	Other [%]	Iter. [-]	Resid. [-]	Sm. r. [-]	Jac. [-]	Int. cycle [s]	Jac. update [h]
Integration to convergence	472.4	100	99.2	0.1	0.2	0.0	0.5	69271	69301	58	2	24.3	0.50
Single shooting, excl. $\underline{P}$, $\underline{V}$	4.4	0.9	34.1	8.8	61.1	0.0	0.1	196	226	58	2	24.1	1.36
Single shooting	7.0	1.5	22.2	5.6	84.3	0.0	0.0	201	231	58	2	24.4	2.97
Multiple shooting, 2 int.	12.8	2.7	14.6	3.3	88.9	0.0	0.1	215	265	59	3	25.4	3.79
Multiple shooting, 4 int.	16.6	3.5	12.8	2.8	84.3	0.1	0.1	227	284	59	3	27.1	4.66
Multiple shooting, 8 int.	32.3	6.8	10.4	1.8	87.6	0.2	0.1	316	352	60	4	34.3	7.08
Multiple shooting, 16 int.	84.0	17.8	6.4	1.0	92.3	0.2	0.0	307	345	63	7	55.0	11.08

Table 12. Results from sequential tests of integration to convergence and the shooting methods. The profiles show: The percentages of the running time spent calculating residuals, performing smoothing runs, computing Jacobians, doing linear algebra on the shooting Jacobians, and the time spent on other tasks. The columns under the heading *Evaluations* show: The number of iterations, residual evaluations, smoothing runs, and Jacobians needed by the methods to complete the test job. The rightmost columns show: The average time needed per evaluation when integrating the entire cycle once and when updating the Jacobian matrix.

Table 12 shows that the shooting methods were significantly faster than integration to convergence. The fastest sequential method was the single shooting method where the dimension of the BVP had been reduced by excluding the variables for pressures and velocities from the shooting.

The profile in Table 12 shows that more than 99 % of the CPU time was spent on updating residuals when integration to convergence was performed. For the shooting methods the time spent computing Jacobians was more significant. Jacobian updates accounted for approximately 61 % of the total running time for the single shooting method with reduced dimension of the BVP, and it increased with N_{ip} to approximately 92 % for the multiple shooting method with 16 sub-intervals in the cycle. The time spent performing linear algebra on the Jacobians for the shooting methods was not significant.

Table 12 shows that the numbers of needed iterations and Jacobians increased with N_{ip} and that the evaluations became significantly more costly as N_{ip} was increased. Jacobian updates typically occurred when changing the working gas or when a new solution had a mean pressure which was significantly different from the last solution.

These results differ from the observations made in the tests where the shooting methods were applied to a model with a simplified momentum balance. For the model with a simplified momentum balance the speed of convergence did not appear to depend on N_{ip} even when as many as 500 sub-intervals were used, and the overhead from splitting the cycle into multiple sub-intervals was negligible for less than 20 intervals in the cycle.

The differences between the results for the two types of models indicate that multiple shooting methods are less attractive for models which include the full momentum balance.

When integrations were started from imperfect initial pressure distributions, as could occur for every sub-interval in the cycle, then pressure transients occurred. For the model with the full momentum balance waves would bounce back and forth until they were finally dissipated. The artificial dissipation described in section 3.2.5 accelerated the dissipation of acoustic waves but the waves still required some time to dissipate completely. As long as acoustic waves were present the IVP solver took very small steps. The number of such transients increased with N_{ip} and this was the main reason for the increase in the computational effort required to perform integrations.

When the sub-intervals of the cycle became short enough the acoustic waves did not have sufficient time to dissipate completely during the integrations. The Jacobians for the multiple shooting method were then polluted by noise which was due to the decaying waves. This noise then masked the true interdependence between the variables and residuals in (24) for periodic steady state solutions. This was believed to be the main reason for the slower convergence observed as N_{ip} was increased. As an experiment N_{ip} was increased to 31 and with this discretisation convergence was difficult to achieve.

Comparison of parallel methods

The absolute speedups from the parallel tests of the shooting method on the model with a full momentum balance are shown in Table 13:

	Speedup: Integrating cycle					Speedup: Updating Jacobians					Speedup: Test job				
	N worker processes					N worker processes					N worker processes				
Method	2	4	8	16	32	2	4	8	16	32	2	4	8	16	32
Single shooting, excl. $\underline{P}$, $\underline{V}$	1.0	1.0	1.0	1.0	0.9	2.0	4.0	7.9	15.7	29.1	1.4	1.8	2.0	2.2	2.2
Single shooting	1.0	1.0	1.0	1.0	1.0	2.0	4.0	8.0	15.4	30.3	1.6	2.3	2.9	3.3	3.7
Multiple shooting, 2 int.	2.0	2.0	1.9	2.0	2.0	2.0	4.0	7.9	15.9	31.3	2.0	3.4	5.2	7.2	8.9
Multiple shooting, 4 int.	1.9	3.5	3.5	3.5	3.5	2.0	4.0	7.9	16.0	31.5	2.0	3.9	6.6	10.1	13.7
Multiple shooting, 8 int.	1.9	3.6	6.0	6.1	6.1	2.0	4.0	7.9	15.9	31.6	2.0	3.9	7.5	13.0	20.1
Multiple shooting, 16 int.	2.0	3.6	5.8	7.3	7.2	2.0	4.0	8.0	15.9	31.1	2.0	4.0	7.7	14.1	23.0

Table 13. Absolute speedups for integrating entire cycles, for computing Jacobians, and for the complete test job for each of the shooting methods.

Table 13 shows that the parallel updates of $\underline{\underline{J_s}}$ and $\underline{\underline{J_{ms}}}$ achieved near linear speedups. This shows that the communications overhead was small and that good load balancing was achieved.

The integrations of the cycle achieved lesser speedups. The best possible speedup for an integration of the cycle was $\min\left(N_{ip}+1, N_{worker\ processes}\right)$ if perfect load balancing was achieved. Significantly smaller speedups were achieved for $N_{ip} > 1$ on more than two processors. The limited speedups were mainly due to imperfect load balancing. The imperfect load balancing was caused by the fast transients in the beginnings of the sub-intervals. The fast transients caused different amounts of overhead in the different sub-intervals.

Table 13 also shows that the absolute overall speedups achieved by the methods for the test job improved with N_{ip}. One reason for this is that the speedups achieved for integrations of the cycle were higher for larger N_{ip}. Another reason was that computations of Jacobians required a larger fraction of the total running time when N_{ip} was increased.

The speedups relative to the sequential running time of the single shooting method, where the pressure and velocities were excluded from the shooting, are shown in Fig. 44:

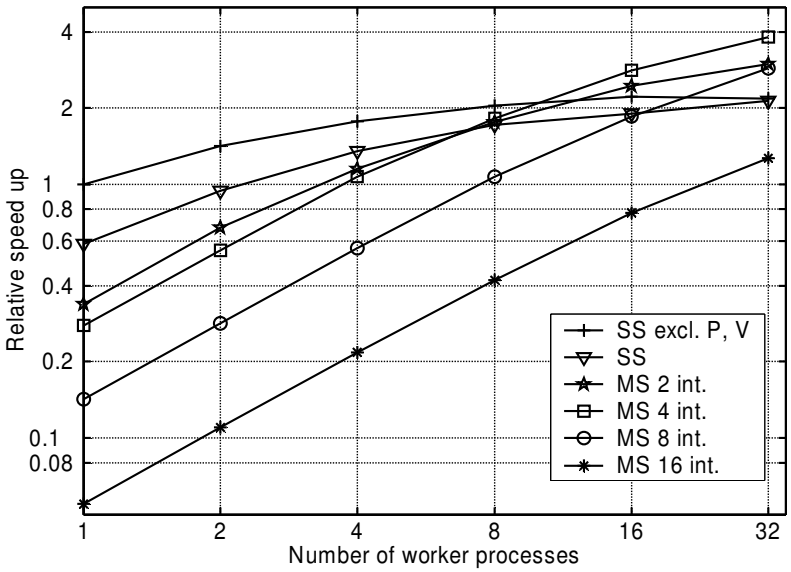


Fig. 44: Speedups relative to sequential single shooting with reduced dimension of BVP achieved by the shooting methods for the model with the full momentum balance.

Fig. 44 shows how the increases in the absolute speedup achieved by increasing N_{ip} for the shooting methods were neutralised by the overhead caused by increasing N_{ip}. The multiple shooting method with $N_{ip} = 3$ on 33 processors achieved the largest relative speedup of all the tested methods, and this relative speedup was only 3.8 for the test job.

4.6.3. Conclusions

The tests where the shooting methods were applied to a batch of 28 solutions to a model with a full momentum balance showed that:

- The single shooting method, where the variables for pressures and velocities were excluded from the shooting, was the fastest of the tested sequential methods. This method also had the fastest convergence. It required less than 1 % of the computational efforts needed to complete the test job with the integration to convergence approach.
- There was a significant overhead from dividing the cycle into sub-intervals. When the cycle was split into 16 sub-intervals the average time to integrate the entire cycle was more than doubled compared to single shooting; the time to update a Jacobian matrix was more than tripled. The convergence also became slower when the number of sub-intervals was increased.

- The absolute scalabilities of the shooting methods on a parallel computer improved with the number of sub-intervals in the cycle. For the batch job of 28 simulations the largest absolute speedup of 23 on 33 processors was observed for the multiple shooting method with 16 sub-intervals in the cycle. The absolute speedups were mainly limited by load balancing issues when integrating the entire cycle.
- Due to the overhead in the shooting methods caused by increasing the number of sub-intervals the best achieved speedup relative to the fastest sequential method was only 3.8. This speedup was achieved for multiple shooting with 4 sub-intervals in the cycle using 33 processors.

4.7. Conclusions for single and multiple shooting methods

The following overall conclusions were drawn from the tests of the single and multiple shooting methods:

- The shooting methods appeared to be reliable for finding periodic steady state solutions to Stirling machine models.
- The multiple shooting methods could be attractive for models where perturbations of initial values do not lead to computationally expensive transients. It is also important that the fast transients are short lived relative to the lengths of the sub-intervals in the cycle, so that the transients do not cause pollution of Jacobians.
- For models built using the modelling approach presented in chapter 3, i.e. for models which include the inertia of the gas in the momentum balance, the single shooting methods with appropriate exclusion of fast variables appear to be the most attractive of the tested methods. Multiple shooting appears to be less attractive unless a large number of processors are freely available.
- For the single shooting methods it was only the updating of shooting Jacobians which achieved a parallel speedup. The parallel computation of the Jacobian made a significant difference for the time needed to produce the first solution. But the overall speedup decreased when the same Jacobian could be recycled for many iterations and solutions during a batch job.

4.8. Parallelisation of shooting batch jobs

Batches of independent simulations are common in practical use of simulation models. Examples of situations where such batch jobs can be found are:

- Parameter studies where the influences of one or more parameters are mapped by changing the parameter(s) in small steps.
- Comparison of sets of simulation results with sets of experimental data.
- Design optimisation. For a conjugate gradients optimisation method, for instance, multiple independent simulations are needed to update the gradient vector of the optimisation method (Numerical Recipes, 1997). If a multipoint approximation method is used then batches of new designs are simulated to update the approximate response functions (Toropov, 2006).

Because such batches of independent simulations occur frequently it has been investigated if the parallel scalability of the single shooting method could be improved for such batch jobs.

The straight forward way to achieve a speedup for a batch job on a parallel computer is to simply run the individual jobs from the batch on different processors. However, shooting jobs which require Jacobians to be computed require a far more CPU time than shooting jobs which do not require a Jacobian to be computed. In practice the long running shooting jobs would dictate the lowest obtainable time required to complete a batch job using this approach.

To achieve better scalability a new parallel shooting method was devised where the parallel shooting methods described above were integrated into a system for performing batch jobs in parallel. The method was created to increase the scalability of the single shooting method, but the implementation was made so that it also works for the multiple shooting methods.

4.8.1. Splitting of batch jobs into sub batches

Because the method will shoot for several solutions simultaneously it is necessary to give some consideration to which jobs should share Jacobians and initial guesses.

When performing optimisation many solutions must be found for only slightly different design parameters. In practice this means that one should always attempt to recycle the newest shooting Jacobian. The Jacobian will then be updated when/if the design optimisation changes the design parameters sufficiently to make the current shooting Jacobian obsolete.

But if a batch job contains solutions corresponding to very different input parameters to a model then it can be difficult to achieve convergence in all the jobs using the same Jacobian. Therefore an option has been added to specify sub batches within a batch job. Jacobians will be shared between the different jobs in the same sub batch, but not between jobs in different sub batches.

Consider the batch of 28 simulations, which was used when validating the Stirling machine model against experimental data for the SM5 engine and which was also used for testing the shooting methods above. This batch job contained simulations with two different working gasses which behaved very differently in the Stirling engine. For this batch job it would be prudent to split the simulations into two sub batches so that each sub batch contained only simulations with one specific working gas.

4.8.2. Parallelisation approach for batch jobs

In section 4.3.6 it was described how the parallelisation of the shooting methods was made using a pool of tasks approach where a single master distributes tasks to a number of worker processes. In order to coordinate concurrent shooting for multiple solutions an additional level of management was added to the parallelisation approach.

The new parallel shooting method for batch jobs uses a single master that controls a group of polymorphic worker processes. The master process can distribute shooting *jobs* to the worker processes and when a worker process receives a job it becomes a *shooter*. When a shooter completes a shooting job it reverts to being a worker.

Performing a shooting job requires tasks (integrations of sub-intervals) to be performed. The master process hosts a shared pool of tasks for all the shooters. The shooters can submit groups of tasks to the master and the master then distributes the tasks among the available worker processes.

Each worker will perform a task, send the results directly to the shooter that submitted the task, and send a short message notifying the master that the task was completed. The message to the master is necessary to let the master know that the worker is available for a new task. Sending the results directly to the shooter that submitted the task, instead of sending the results through the master process, reduces the bandwidth requirement for the master process.

With this new approach there will hence typically be 3 types of processes coexisting simultaneously: A *master*, some *shooters*, and some *worker*s.

Integrations of the cycle

When a shooter needs the entire cycle to be integrated it can either submit the integration(s) as a group of tasks to the master or it can perform the integration(s) itself.

If single shooting is performed, so that there is only one sub-interval in the cycle, then it will usually be optimal if the shooter performs the integration itself, instead of submitting it to the master. The shooters will then spend no time waiting for results, no communications overhead is induced, and no parallelism is lost. If multiple shooting is performed then a speedup can be achieved by integrating the sub-intervals in the cycle in parallel and so the integrations should be submitted to the master. The exact behaviour can be customised via settings passed to the shooting method.

The master will always accept requests for simulations of the entire cycle and the corresponding tasks will be added to the shared pool of tasks.

Jacobian updates

When a shooter needs an update of its Jacobian then it will check if a newer Jacobian than the one currently in memory has been stored in a file on disk. If it finds a newer Jacobian in a file it will load the new Jacobian and continue its iterations with the newer Jacobian.

If no newer Jacobian is found on disk then the shooter will submit a request for a Jacobian update to the master. The master will check if a Jacobian update is already in progress within the same sub batch. If no Jacobian update is in progress in the same sub batch then the master will accept the group of tasks corresponding to the Jacobian update and add it to the shared pool of tasks. If, however, a Jacobian update is in progress in the same sub batch then the shooter will be enqueued to wait for the completion of that Jacobian update.

When a shooter is waiting for another shooter to complete a Jacobian update it will act as a worker and will be treated as such by the master, with one exception: A shooter that is acting as a worker while waiting for a Jacobian update will not receive any new jobs, only tasks. When a Jacobian update is completed the Jacobian will be stored in a file, and then all shooters that are waiting for the Jacobian are released. They then read the new Jacobian from the file and continue their iterations.

Task prioritisation and performance issues

The master generally loops through the task buffer in a cyclic fashion so that all shooters are served successively in a round robin manner. This behaviour can be customised through the settings of the method so that, for instance, the master can be forced to concentrate all workers on finishing one Jacobian update at a time instead of working simultaneously on multiple Jacobian updates for different sub batches.

The number of concurrent shooters can be important to performance.

When a shooter submits a group of tasks to the master and the group of tasks is accepted then the shooter will begin to wait for the results from the tasks. While waiting and receiving results the shooter does not fully utilise a CPU for performing useful work. Having a large number of shooters running simultaneously can thus result in wasted CPU resources. Sometimes this problem can be eliminated by running multiple processes pr. processor and letting the operating system balance the load. But some MPI implementations spend excessive amounts of CPU resources when waiting for messages in order to keep processes in as high a state of readiness as possible. This is done to reduce the latency for receiving messages but it has the side effect that running multiple processes pr. processor becomes impractical.

If on the other hand the number of shooters is small then there is a risk that the flow of submitted tasks from the shooters is too moderate to keep all the workers busy. One can thus expect that an optimum number of shooters exist for a given batch job.

4.8.3. Handling failed simulations

Robustness is very important when simulation programs are used in conjunction with numerical optimisation algorithms. Because of this it is necessary to have methods for handling failed simulations in the shooting methods. A failed simulation can occur if a simulation is started with a very poor guess for the initial values and shooting parameters. This can occur, for instance, if the shooting diverges because the Jacobian is not sufficiently up to date.

Failed simulations are most likely to occur during the updates of the residuals that take place in the iterations of the shooting method. This is because the residual updates are performed when the initial values and shooting parameter have been updated as shown in (23), (25), or (27); it is hence in the residual updates that a divergence would first be observed.

Simulations during an update of the Jacobian matrix are less likely to fail. Before a Jacobian update is performed the residuals have already been calculated at the current guess for the solution. The initial values and shooting parameter values used during Jacobian updates differ from the current guess for the solution only by the small perturbations applied in the numerical differencing when calculating the elements of the Jacobian.

If a group of tasks contain tasks that cause failed simulations then the following series of events illustrate the general scheme used to recover from the failure(s):

1. When a task fails then the worker, which was trying to complete the task, sends a results message containing a raised error flag to the shooter that submitted the task to the master. This causes the shooter to listen for a message directly from the master. The same worker then sends a message to the master that the task caused a failed simulation.
2. Notification of a failed task causes the master to send a message to the listening shooter specifying how many of the tasks in current group of tasks have already been dispatched to workers. The master then removes any remaining tasks in the current group of tasks from the task buffer so that no more of the tasks will be dispatched.
3. The shooter continues to receive results until the results from all the tasks, which had already been dispatched by the master, have been received.
4. The shooter will then act on the failure.

The handling of failed simulations is illustrated in the Fig. 45.

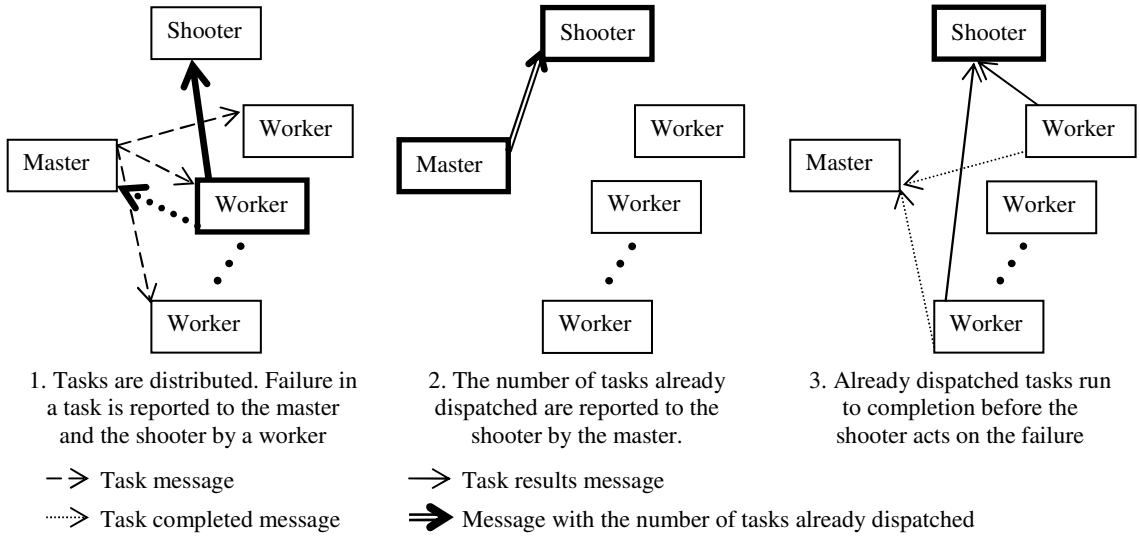


Fig. 45. Illustration of the general scheme for recovering from failed simulations when performing multiple integrations in parallel.

If the group of tasks, which contained the failed task, was an integration of the entire cycle, then the subroutine that submitted the tasks will exit with an error flag when the results from all dispatched tasks have been received. It is then the shooting method itself that, if possible, tries to recover from the failed simulation. The shooting method can try to recover by going back to the best known guess for the solution since the last Jacobian update and:

1. Try to apply only a fraction of the current corrections.
2. Request a new Jacobian which can be used for computing a new and better estimate for the needed corrections.

When working on batch jobs the shooting method can actually also attempt to recover from a failed simulation in a Jacobian update. If a Jacobian update fails and if another shooters within the same sub batch is waiting for the Jacobian, then the shooters can swap roles. The shooter which was waiting for the failed Jacobian update will then try to have its own Jacobian updated. If the new Jacobian update is successful then the shooter whose Jacobian update failed can load the successfully updated Jacobian instead.

4.8.4. Dynamic IVP solver tolerance management

It has been observed that the speed of convergence for the shooting methods, when applied to models of Stirling machines and pulse tube coolers, is usually not larger than a reduction of one or two orders of magnitudes for the residuals in each iteration. In the results presented in Table 12 the fastest converging method, single shooting with pressures and velocities excluded, required seven iterations on average. This observation indicates that the IVP method probably integrated using a stricter tolerance than needed in the early iterations. Almost certainly there was little benefit from using a

strict tolerance in the smoothing runs performed at the beginning of the shooting for each new solution.

As shown in Fig. 18 it can sometimes, but not always, save computational effort to use a less strict tolerance during integrations of a model. It is thus plausible that a speedup can sometimes be achieved by managing the tolerance applied in the IVP method dynamically.

The strategy attempted here has been to estimate the needed tolerance for the IVP method as a fraction of the size of the maximum relative residual corresponding to the current guess for the solution. This strategy was applied when updating residuals and performing smoothing runs. A strict tolerance was always used when performing integrations for updates of the Jacobian matrix for the shooting method.

4.9. Tests of the shooting method for batch jobs

The new implementation of the shooting method for batch jobs was tested with the same batch job and on the same platform as described in section 4.6.

4.9.1. Method

The following configuration appeared reasonable for this batch of 28 simulations and was used in the tests:

- The shooters used single shooting where the variables for pressures and velocities were excluded from the shooting.
- The shooters performed the integrations of the cycle sequentially.
- No limit was imposed on the number of shooting jobs that the master could distribute simultaneously.
- The tolerance for the IVP method was adjusted by setting it two orders of magnitude smaller than the largest relative residual in the shooting method. The tolerance for the IVP method was limited to vary in the interval from 10^{-5} to 10^{-7}.

I did not succeed in coercing the MPI implementation and operating system on the test platform into running multiple processes pr. processor without significant overhead. Therefore each process occupied one processor in the tests.

With these settings the following events would occur during a typical test run:

- The master would start $\min(N_{jobs}, N_{processors} - 1)$ shooting jobs.
- The shooters would run sequentially until they needed to update their shooting Jacobians.
- Requests for Jacobian updates were processed by the master. Only one Jacobian update pr. sub batch was allowed to be in progress at any one time.
- Shooters that submitted requests for Jacobian updates which were not accepted worked together with any worker processes on completing the first Jacobian(s). Only the shooter(s) of the accepted Jacobian update(s) remained lightly loaded during the update(s).
- When the first Jacobian(s) were completed the shooters continued their iterations using the new Jacobians.

- When the shooters finished their jobs they would receive new jobs if any jobs remained in the batch, and otherwise they would remain available as workers.
- As the batch job neared completion the last few shooters would be iterating to finish their last jobs. The remaining worker processes would remain idle until the completion of the batch job unless a Jacobian update was needed.

4.9.2. Results and discussion for the shooting method for batch jobs

The speedups measured in the tests of the new parallel shooting method for batch jobs are shown in Fig. 46. The new method is denoted *MSS excl. P.V.* (multiple-single-shooting with pressures and velocities omitted from the shooting process) in the figure.

When run sequentially the new method required 2 hours and 45 minutes on one of the 1.05 GHz processors on the Sun Fire 15000. This was 37 % faster than the results presented in section 4.6.2 for the single shooting method with pressures and velocities excluded. This difference was due to the dynamic management of the tolerance of the IVP method.

The relative speedups for the previously tested shooting methods, i.e. the results presented in Fig. 44, have been recalculated to reflect the new fastest sequential method and have been included in the figure for reference.

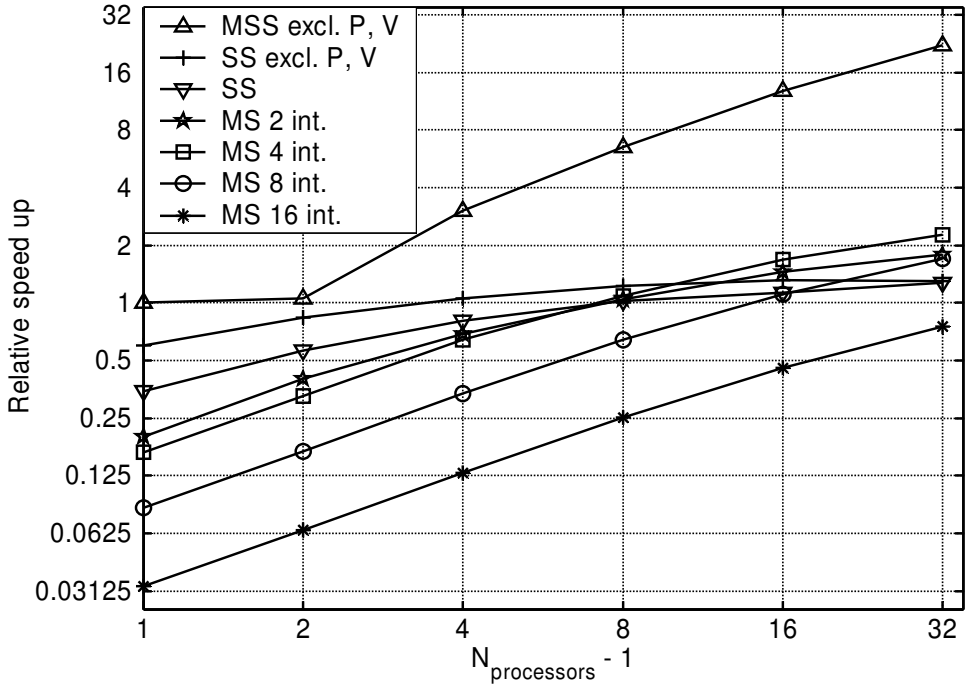


Fig. 46: Speed up of the new shooting method for batch jobs measured during benchmarking on a Sun Fire 15000 server. The results from Fig. 44 have been recalculated to reflect the new fastest sequential method and have been included in the figure.

When run with two processes, i.e. with one shooter and one master which each occupied one processor, the performance of the new method was identical to the sequential performance of the new method.

When three processors were used there was one master and two shooters. In this configuration the shooters had to wait on each other during Jacobian updates. This meant that one of the shooters would act as a worker and would perform the integrations needed for the Jacobian update on the other shooter. This meant that there was no speedup for Jacobian updates even though both shooters would be tied up when performing the Jacobian update. This was the reason for the poor speedup on 3 processors.

When more than 3 processors were used the new method scaled well. A speedup of 22 was achieved for the batch job when the method was run on 33 processors. In this case the batch of 28 simulations was completed in approximately 7.5 minutes.

In a sequential run with the new implementation an average of seven iterations pr. solution was needed. This is the same number of iterations as in the results presented in section 4.6.2 and this shows that the dynamic management of the tolerance of the IVP method did not significantly affect the speed of convergence for the shooting method.

When jobs had been completed the shooters would use the solutions from the completed jobs as initial guesses for new jobs. Because the order in which initial guesses became available depended on the number of processors the numbers of iterations also depended slightly on the number of processors. The maximum average needed number of iterations observed in the tests was 7.8.

Two Jacobians, one for helium and one for nitrogen as working gas, were needed to complete the batch job. This was independent of the number of processors.

4.9.3. Conclusion

The implementation for batch jobs of the single shooting method with suitable exclusion of variables was the fastest and most reliable method tested in this work.

4.10. Summary

Single- and multiple shooting methods for finding periodic steady state solutions have been studied. The study showed that the shooting methods appeared to be fast and reliable for finding accurate periodic steady state solutions.

It was found that the single shooting method had a significantly lower computational cost pr. solution than multiple shooting methods when applied to models built using the modelling approach from chapter 3. This was primarily because the cost of integrating the cycle and the cost of updating Jacobians increased significantly when the number of sub divisions of the cycle was increased. The overhead for the multiple shooting method was found to be specific to models which included the inertia of the gas in the momentum balance for the gas. For a model where the momentum balance had been simplified by excluding the inertia of the gas the multiple shooting method was more competitive.

The computational cost of the single shooting method could be reduced by excluding the fastest variables, such as pressure and velocities, from the shooting. Fixed point iteration was then performed on the excluded variables.

The performance could be further increased by letting the shooting method manage the tolerance used by the IVP solver when integrating the model. In this way a

less strict tolerance could be used until the shooting method was close to the true periodic steady state solution.

The shooting methods were parallelised on a coarse grained level using MPI. The parallelisation was first made using a simple pool of tasks approach. The parallel speedups for the shooting methods were measured on a batch job with 28 solutions which covered a wide range of operating conditions for a model of the SM5 Stirling engine.

With respect to absolute speedups for the shooting methods it was found, that the updates of Jacobians for both the single and multiple shooting methods scaled well. Absolute speedup factors between 29.1 and 31.6 were achieved on 33 processors for 1 to 16 sub-intervals in the cycle. Jacobian updates scaled better for the multiple shooting methods because they facilitated better load balancing.

Integrations of the cycle, which were performed to update residuals or as smoothing runs, achieved absolute speedups up to 7.2 on 33 processors for the multiple shooting method with 16 sub intervals in the cycle. It was found that load balancing issues limited the speedup for integrations of the cycle. The single shooting method achieved no parallel speedup for integrations of the cycle.

When computing one solution at a time the absolute speedups which could be achieved by the shooting methods for a given batch job hence depended on the mix between Jacobian updates and iterations. When most of the time was spent on updating Jacobians the parallel speedup would be very good. But if Jacobians could be recycled for many iterations then the parallel speedup would decrease. For the batch of 28 simulations the achieved absolute speedups on 33 processors varied from 2.2, for the single shooting method with pressures and velocities excluded, and up to 23 for the multiple shooting method with the cycle divided into 16 sub-intervals.

When the relative speedups of the methods were studied it was found that the larger computational efforts required to perform multiple shooting instead of single shooting almost neutralised the advantages of multiple shooting with respect to absolute scalability. The largest speedup relative to the sequential performance of the single shooting method with pressures and velocities excluded was only 3.8 on 33 processors for the batch of 28 simulations. This speed up was achieved for multiple shooting with four sub-intervals in the cycle.

The implementation of the shooting methods was refined so that it could use the parallelism inherent in batch jobs for improving the scalability of the parallel shooting methods for batch jobs. The new implementation was also based on a pool of tasks approach and it was designed for robustness, so that the parallel shooting methods were suitable to be used in conjunction with a numerical optimisation algorithm.

The new implementation enabled the single shooting method with pressures and velocities excluded to achieve a speedup of 22 on 33 processors for the batch of 28 simulations. This method was the fastest of all the tested methods.

(This page intentionally left blank)

5. Numerical studies on phenomena in Stirling machines and pulse tube coolers

During this work the developed simulation tools have been used for performing a number of numerical studies on different phenomena in Stirling machines and pulse tube coolers. Results from these studies have been published (or submitted for publication) in journals and at conferences. Four of these papers accompany this thesis as self contained studies. Brief introductions to the papers are given below.

5.1. Preliminary Results from Simulations of Temperature Fluctuations in Stirling Engine Regenerator Matrices (Paper A)

Results from a study on the differences caused by using, respectively, constant matrix temperatures and oscillating matrix temperatures when modelling the SM5 Stirling engine were published in this paper.

The study was performed using a model which contained a number of simplifications compared to the Stirling machine model described in section 3.4. In particular the model used for this study neglected the inertia of the gas. It was this simplified model which was used for the tests of the shooting method described in section 4.5.

In the study the SM5 engine was simulated with both constant and oscillating matrix temperatures, and the influence of the spatial discretisation was studied for both cases. It was also documented that the periodic steady state solutions, which had been calculated for the study using a shooting method, were true steady state solutions.

In the study it was found that the axial temperature profile in the regenerator matrix looked significantly different when matrix temperature oscillations were taken into account. In the results which included matrix temperature oscillations there were significant matrix temperature oscillations in the ends of the regenerator and a smaller temperature gradient in the central part of the regenerator.

It was also found that the matrix temperature oscillations had a significant impact on the performance of the SM5 engine and that the matrix temperature oscillations therefore deserved further study.

5.2. Numerical study on optimal Stirling engine regenerator matrix designs taking into account the effects of matrix temperature oscillations (Paper B)

This paper presented results from a study on the effects of regenerator matrix temperature oscillations on the performance of the SM5 Stirling engine for a wide range of regenerator matrix thread diameters and porosities.

In this study a model similar to the Stirling machine model described in section 3.4 was used for simulating the SM5 engine. This model included the inertia of the gas in the momentum balance equations. In the study the heat transfer and flow friction in the regenerator was described using empirical correlations for metal felt regenerators.

The performance of the SM5 engine was mapped for wire diameters between 15 and 150 μm and fill factors between 0.01 and 0.27 (equivalent to void fractions between

0.99 and 0.73). This mapping was performed for both oscillating and constant matrix temperatures. The influence of the matrix temperature oscillations was isolated by subtracting the results for constant matrix temperatures from the results for oscillating matrix temperatures.

From the numerical results it was observed that the regenerator matrix temperatures appeared to oscillate in two modes. The first mode was oscillation of a nearly linear axial matrix temperature profile while the second mode bended the ends of the axial matrix temperature profile when gas flowed into the regenerator with a temperature significantly different from the matrix temperature. The first mode of oscillation improved the efficiency of the engine but the second mode reduced both the work output and efficiency of the engine. The magnitudes of the different modes of oscillation depended on the matrix design.

The results showed that the efficiency of the SM5 Stirling engine could be slightly improved by using a new regenerator with three differently designed matrix sections. The new matrix design amplified the first mode of oscillation and reduced the second mode.

5.3. Numerical study on transverse asymmetry in the temperature profile of a regenerator in a pulse tube cooler (Paper C)

This paper documented the findings of a numerical study on transverse asymmetry in the temperature distribution of a regenerator matrix in a pulse tube cooler. The study was performed to identify the mechanism which had caused a transverse asymmetry in the temperature distribution of a geometrically symmetric regenerator in a PTC prototype. It had been observed in experiments that the performance of the PTC was reduced when the asymmetry was present.

The study reproduced the experimentally observed transverse asymmetry using the PTC model which was validated in section 3.9.3. The model was built so that the regenerator could either be modelled as a single channel or be divided into two parallel regenerator channels, each with half the cross sectional area of the single regenerator channel. To reproduce the asymmetry the regenerator was modelled with two parallel channels and the PTC was simulated as an initial value problem.

It was found that a small perturbation of the symmetric periodic steady state solution was sufficient to trigger an instability that caused the axial temperature profiles in the two regenerator channels to diverge and become very different. It was found that the asymmetry was caused by a circulating flow in the closed circuit formed by the two parallel regenerator channels and the manifolds at the ends of the regenerator. The circulating flow was superimposed on the oscillating axial flow through the regenerator. The asymmetry and the circulating flow increased the regenerator loss by up to a factor of five in the simulations.

A separate simple model of a single regenerator channel was used to identify the mechanism driving the circulating flow. In the simple model the entire regenerator channel was lumped into a single control volume. The dependence of the net flow rate through the regenerator channel on the average matrix temperature, matrix temperature

oscillations, and oscillations in the average pressure was studied for various wave forms for the pressure difference between the ends of the regenerator channel.

The primary mechanism driving the circulating flow was found to be due to the shape of the pressure difference wave, and the dependence of the instantaneous mass flow rate on the instantaneous pressure difference between the ends of the regenerator and on the temperature in the regenerator. A smaller contribution to the circulating flow was linked to the temperature oscillations in the regenerator. These mechanisms were found to increase the tendency of a regenerator channel to draw in mass from the cold end of the regenerator if the temperature in the channel decreased, and conversely to make a hot regenerator channel draw in more mass from the hot end of the regenerator. These mechanisms hence induced a circulating flow that amplified transverse asymmetry in the regenerator matrix temperature profiles. A weaker mechanism due to the oscillations in pressure was found to have the opposite effect and hence worked against the transverse asymmetry.

5.4. Numerical study on the appendix gap losses in a Stirling engine (Paper D)

This paper reported on a numerical study on the appendix gap losses in the SM5 Stirling engine.

The study was performed using the Stirling machine model described in section 3.4 which was validated against experimental data for the SM5 engine in section 3.9.1. The appendix gap is included directly in the Stirling machine model where both the displacer wall, the cylinder wall, and the gas in the gap are discretised. Because the displacer is moving relative to the cylinder wall the control volumes containing the gas in the gap slide over the control masses used for discretising the cylinder wall. The technicalities related to the moving mesh are discussed in the paper.

The influence of the spatial discretisation on the calculated appendix gap losses was studied. Parametric studies on the size of the gap were then performed using two different methods for computing the heat transfer in the gap. One method for computing the heat transfer in the gap was to use simple polynomial approximations for the radial temperature profiles in the gap. The second method for computing the heat transfer in the gap was to use empirical correlations from the literature. The results from the parametric studies were compared to analytically derived formulas for the appendix gap losses.

The study on the convergence of the spatial discretisation in the gap showed that a relatively coarse discretisation appeared to be adequate for studying the appendix gap losses. The wall temperature profiles calculated during the study also showed that the axial gradients in the wall temperatures varied significantly along the gap.

The parametric studies on the gap size yielded larger appendix gap losses for small gap sizes but smaller losses for large gap sizes compared to the analytically derived formulas for the appendix gap losses. The larger losses for smaller gap sizes were attributed to a steep temperature gradient on the cylinder wall at the location of the regenerator. The smaller appendix gap losses for large gaps occurred because the heat transfer in the gap was less effective when the hydraulic diameter of the gap was large; this effect was neglected in the analytically derived formulas.

The calculations where empirical correlations for the heat transfer in the gap were used yielded lager appendix gap losses than the calculations where the heat transfer was calculated from approximate polynomial temperature profiles. But the gap sizes where local extrema in the appendix gap losses occurred were very similar for the two methods for computing heat transfer in the gap.

The impact of the appendix gap losses on the heat intake and work output of the engine was also studied. A close correspondence was found between the appendix gap losses and an increased heat intake of the engine. The appendix gap losses were also found to reduce the work output of the engine; especially for relatively large gaps where the pumping loss dominated the appendix gap losses. The results indicated that the appendix gap losses in the SM5 engine could be reduced by making the gap larger than in the present design.

6. Summary and perspective

A simulation approach, where models are formulated using a control volume based modelling approach for one-dimensional compressible flow and where the shooting method is used for finding periodic steady state solutions, has been developed. The simulation approach represents a compromise between model complexity and accuracy on one side, and practical waiting times for results on the other.

The modelling approach has been implemented in a way that generates a model source file where individual equations can be accessed and modified. This makes the models very flexible and adaptable. This flexibility has been an asset when performing the numerical studies reported on in this work.

The consistency, accuracy, and computational performance of models built using the developed modelling approach have been studied. The abilities of the models to predict the performance of actual Stirling machines and pulse tube coolers have also been successfully tested against both experimental data and results from current state of the art software. The models have been used for numerical studies on phenomena and loss mechanisms in Stirling machines and pulse tube coolers.

Significant efforts were put into finding fast numerical algorithms for the simulations because good numerical efficiency means that more details can be included in models without making simulations impractically slow. Parallel programming has also been used as a method for speeding up IVP solving and calculations of periodic steady state solutions.

The time needed to find periodic steady state solutions depends on a number of factors including the model, the simulated machine and its operating conditions, and the chosen discretisation. For the models developed in this work it was possible to choose the discretisations so that numerical optimisations of complete machine designs were practical on a PC.

The methods and tools developed in this work appear to function well, but significant amounts of further work and development is necessary if the tools are to become fully accessible and useful to users other than their developer.

When this work began parallel computers were probably considered to be somewhat exotic. But today “parallel computers” are becoming widespread in PCs and laptops through the advent of inexpensive dual-core processors. Quad-core processors for PCs have already been showcased. The trend towards multi-core processors is backed by the company Intel whose vision for the future of PC processors includes processors with “...many tens of cores, potentially even hundreds of cores...” (Intel, 2005). The trend towards multi-core processors has made the efforts expended on parallelisation in this work less exotic because parallel algorithms are necessary to fully utilize the potential of new PCs and notebook computers.

(This page intentionally left blank)

References

Adewumi M A. Eltohami, E S. Ahmed, W H. 2000. *Pressure Transients Across Constrictions.* Journal of Energy Resources Technology (122), ASME, pp. 34-41.

Alexander, R. 2003. *Design and implementation of DIRK integrators for stiff systems*, Applied Numerical Mathematics (46), Elsevier, pp. 1-17.

Altman, A. 2003. *SNAPpro Stirling Numerical Analysis Progam*, In: Proc. of the 11th International Stirling Engine Conference, Rome, University of Rome "La Sapienza", 2003, pp. 166-172.

Altman, A. 2005. SNAPpro web page, http://home.comcast.net/~snapburner/, visited on 21/10-2005.

Andersen, A. 2003. *Metanisering af CO og CO_2.* Masters thesis project at Department of Chemical Engineering, Technical University of Denmark, January 2003.

Andersen, S K. Carlsen, H. Thomsen, P G. 2004. *Control volume based modelling of compressible flow in reciprocating machines*. In: Proc. of the 45th International Conference of the Scandinavian Simulation Society (SIMS2004), Copenhagen, 2004, pp. 119-128. NOTE: The paper was accepted for publication on 17/6-2005 for a SIMS 2004 special issue of Simulation Modelling Practice and Theory, Elsevier. Here the paper will appear with the title *Control volume based modelling in one space dimension of oscillating, compressible flow in reciprocating machines*.

Ash, J E. Heames, T J. 1981. *Comparative Analysis Of Computer Codes For Stirling Engine Cycles*. In: Proc. of the 16th Intersociety Energy Conversion Engineering Conference, Atlanta Georgia, ASME, 1981, pp. 1936-1941.

Bauwens, L. 1990. *Consistency, Stability, Convergence of Stirling Engine Models*. In: Proc. of the 25th Intersociety Energy Conversion Engineering Conference, Reno Nevada, American Institute of Chemical Engineers, 1990, pp. 352-358.

Bauwens, L. 1993a. *Stirling Engine Modelling: The MS*2 Code*. In: Proc. of the 6th International Stirling Engine Conference, Eindhoven, 1993, pp. 371-376.

Bauwens, L. 1993b. *Stirling Engine Modelling on Eulerian Grids*. In: Proc. of the 28th Intersociety Energy Conversion Engineering Conference, Atlanta Georgia, American Chemical Society, 1993, pp. 2725-2730.

Bauwens, L. 1996. *Heating At The Closed End Of A Duct Due To Oscillating Flow - The Transient Regime.* In: Proc. of the 31th Intersociety Energy Conversion Engineering Conference, Washington DC, Institute of Electrical and Electronics Engineers, 1996, pp. 1212-1217.

Bird, R. Stewart, W. Lighfoot, E. 1960. *Transport Phenomena*. John Wiley & Sons, ISBN 0-471-07392-X.

Cairelli, J E. Geng, S M. Skupinski, R C. 1989. *Results From Baseline Tests Of The SPRE I And Comparison With Code Model Predictions.* In: Proc. of the 24th Intersociety Energy Conversion Engineering Conference, Los Alamitos, CA, IEEE Publications, 1989, pp. 2249-2256.

Cairelli, J E. Swec, D M. Skupinski, R C. Rauch, J S. 1990. *Update On Results Of SPRE Testing At NASA.* In: Proc. of the 25th Intersociety Energy Conversion Engineering Conference, Reno Nevada, American Institute of Chemical Engineers, 1990, pp. 237-244.

Calandrelli, L. Rispoli, F. 1995. *Wave Propagation Method For Stirling Engine Cycle Simulation: An Experimental Validation.* In: Proc. of the 30th Intersociety Energy Conversion Engineering Conference, Orlando Florida, ASME, 1995, pp. 397-406.

Carlsen, H. 1993. *10 kW Hermetic Stirling Engine For Stationary Applications.* In: Proc. of the 6th International Stirling Engine Conference, Eindhoven, 1993, pp. 401-406.

Carlsen, H. Bovin J. 2001. *Test of 9 kW Stirling engine using biogas as fuel*, 10th International Stirling Engine Conference, VDI, Osnabrück, September 2001.

Carlsen, H. 2006. Home page with contact information for Professor Henrik Carlsen, Energy Engineering Section, Department of Mechanical Engineering, Technical University of Denmark. http://www.mek.dtu.dk/Medarbejdere/MEK_ET.aspx?lg=showcommon&id=320&type=person, last visited on 16/3-2006.

Chandrashekar, P. 2006. Home page of Praveen Chandrashekar, CTFD Division, National Aerospace Laboratories, INDIA. http://pc.freeshell.org/prog/, last visited on 10/1-2006.

Chow, H S. Mahkamov, K. 2005. Attempting a coupled "Burner-Engine" CFD Simulation. In: Proc. of the 12th International Stirling Engine Conference, Durham UK, Durham University, 2005, ISBN 0-9535-5582-8, pp. 380-388.

Commiso, M B. 1994. *Analysis of Cyclic Thermodynamic Processes*. Ph.D. thesis at Laboratory for Energetics, The Technical University of Denmark, ISBN 87-7475-160-3.

Dietrich, M. Yang, L.W. Thummes, G. 2005. *High-power Stirling-type pulse tube cooler: observation and elimination of regenerator temperature inhomogeneities*, submitted to Cryogenics

Fedkiw, R P. Merriman, B. Donat, R. Osher, S. 1998. *The Penultimate Scheme for Systems of Conservation Laws: Finite Difference ENO with Marquina's Flux Splitting*. Solutions of PDE Conference - in honour of Prof. Roe, Arachaon, France, July 1998.

Gedeon, D. 1984. *Numerical Advenction Errors In Stirling Cycle Nodal Analysis.* In: Proc. of the 19th Intersociety Energy Conversion Engineering Conference, San Francisco California, American Nuclear Society, 1984, pp. 1898-1904.

Gedeon, D. 1986. *A Globally-Implicit Stirling Cycle Simulation.* In: Proc. of the 21st Intersociety Energy Conversion Engineering Conference, San Diego California, American Chemical Society, 1986, pp. 550-554.

Gedeon, D. 1989. *Modelling 2-D Jets Impinging on Stirling Regenerators.* In: Proc. of the 24th Intersociety Energy Conversion Engineering Conference, Los Alamitos, CA, IEEE Publications, 1989, pp. 2199-2203.

Gedeon, D. 1994. Sage: *Object Oriented Software for Stirling Machine Design.* In: Proc. of the 29th Intersociety Energy Conversion Engineering Conference, Monterey California, American Institute of Aeronautics and Astronautics, 1994, pp. 1902-1907.

Gedeon, D. Wood, J G. 1996. *Oscillating-Flow Regenerator Test Rig: Hardware and Theory With Derived Correlations for Screens and Felts*. NASA Contractor Report 198442.

Geng, S M. Tew, R C. 1992. *Comparison of GLIMPS and HFAST Stirling Engine Code Predictions with Experimental Data.* In: Proc. of the 27th Intersociety Energy Conversion Engineering Conference, Warrendale PA, Society of Automotive Engineers, 1992, pp. 553-558.

Gromol, B. Huber, N. Dietrich, M. Yang, L.W. Thummes, G. 2006. *Development of a 25 K Pulse Tube Refrigerator For Future HTS-Series Products in Power Engineering.* Presented at Cryogenic Engineering Conference 2005, Keystone Colorado, paper no. C1-S-01. (to appear in Advances in Cryogenics Engineering, Volume 51, 2006)

Huang. S.C. 1992. *HFAST - A Harmonic Analysis Program for Stirling cycles*. In: Proc. of the 27th Intersociety Energy Conversion Engineering Conference, Warrendale PA, Society of Automotive Engineers, 1992

Huang, S.C. Berggren, R. 1986. *Evaluation of Stirling Engine appendix gap losses*. In: Proc. of the 21st Intersociety Energy Conversion Engineering Conference, San Diego California, American Chemical Society, 1986, pp. 562-568.

Ibrahim, M. Tew, R C. Dudenhoefer, J E. 1989. *Two-Dimensional Simulation Of A Stirling Engine Heat Exchanger.* In: Proc. of the 24th Intersociety Energy Conversion Engineering Conference, Los Alamitos, CA, IEEE Publications, 1989, pp. 2795-2802.

Ibrahim, M. Tew, R C. Dudenhoefer, J E. 1990. *Further Two-Dimensional Code Development for Stirling Space Engine Components.* In: Proc. of the 25th Intersociety Energy Conversion Engineering Conference, Reno Nevada, American Institute of Chemical Engineers, 1990, pp. 329-335.

Ibrahim, M B. Tew, R C. Zhang, Z. Gedeon, D. Simon, T W. 2001. *CFD Modeling of Free-Piston Stirling Engines*. NASA Technical report NASA/TM—2001-211132, 2001

Intel. 2005. *Platform 2015 Unveiled at IDF Spring 2005: Processor and platform evolution for the next decade*. Webpage at:

http://www.intel.com/technology/techresearch/idf/platform-2015-keynote.htm . Last visited 21/2-2006.

Jansson, P. 2004. *Motor Design: Thinking in 3d.* Appliance Design, September 1, 2004. From the webpage: http://www.appliancedesign.com/CDA/Archives/5bacd7f638a38010VgnVCM100000f932a8c0____

Kühl, H D. 1990. *Veralgemeinerte thermodynamishe Bescreibung regenerativer Gaskreisprozesse.* Doctoral thesis. Fortschritt Berichte VDI Reihe 19 Nr. 42, VDI-Verlag GmbH, Düsseldorf, 1990.

Kühl, H-D. Schulz, S. 1995. *Modeling Of Thermal Regenerator Losses Due To Axial Flow Dispersion.* In: Proc. of the 7th International Stirling Engine Conference on Stirling Cycle Machines, Tokyo Japan, Waseda University, 1995, pp. 315-320.

Kühl, H-D. Pfeffer, T. Schulz, S. Walther, C. 1997. *High Speed Gas Temperature Measurements In A Vuilleumier Heat Pump And Their Reproduction By Differential Computer Simulation.* In: Proc. of the 8th International Stirling Engine Conference and Exhibition, Ancona Italy, University of Ancona, 1997, pp. 257-266.

LfBS RWTH Aachen. 2002. Internet download page titled: *MP-MPICH: Multi-Platform MPICH*. Lehrstuhl für Betriebssysteme (LfBS), RWTH Aachen, Kopernikusstr. 16, D-52056 Aachen. http://www.lfbs.rwth-aachen.de/mp-mpich/download/ . Version: MP-MPICH 1.3.0-a.

Lista, P. 1993a. *The GLIMPS simulation code applied to the V160 DMA Stirling cogeneration unit.* In: Proc. of the 28th Intersociety Energy Conversion Engineering Conference, Atlanta Georgia, American Chemical Society, 1993, pp. 2683-2688.

Lista, P. 1993b. *The Martini Weiss simulation code applied to the V160 DMA Stirling cogeneration unit.* In: Proc. of the 28th Intersociety Energy Conversion Engineering Conference, Atlanta Georgia, American Chemical Society, 1993, pp. 2695-2700.

Mahkamov, K. Ingham, D. B. 2000. *Theoretical Investigations On The Stirling Engine Working Process.* In: Proc. of 35th Intersociety Energy Conversion Engineering Conference, Las Vegas Nevada, AIAA, 2000, pp. 101-110.

Mahkamov, K. Djumanov, D. 2003. *Three-dimensional CFD modelling of a Stirling engine.* In: Proc. of the 11th International Stirling Engine Conference, Rome Italy, University of Rome "La Sapienza", 2003, pp. 97-107.

Mahkamov, K. Djumanov, D. Hislop, D W. 2003. *Results Of A Second-Order Analysis And Three-Dimensional CFD Simulations Of "Gamma"- And "Alpha"-Layouts Of A SES Engine For A Biomass Application*. In: Proc. of the 11th International Stirling Engine Conference, Rome Italy, University of Rome "La Sapienza", 2003, pp. 108-119.

Marcus-Moeller, C. Mortensen, H H. 2005. *Theoretical and Experimental Analyses of Regenerators.* Midterm project at the Department of Mechanical Engineering, Energy Engineering Section, Technical University of Denmark, February 2005.

Martini, W R. 1978. *Stirling Engine Design Manual*. DOE/NASA Technical report DOE/NASA/3152-78/1 (NASA CR-135382), Joint Center For Graduate Study, 1978

Martini, W R. 1983. *Stirling Engine Design Manual Second Edition*. DOE/NASA Technical report DOE/NASA/3194-1 (NASA CR-168088 and/or CR-158088), NASA Lewis Research Center, 1983

Martini, W R. 1986. *A Stirling Engine Analysis Method Based Upon Moving Gas Nodes.* In: Proc. of the 21th Intersociety Energy Conversion Engineering Conference, San Diego California, American Chemical Society, 1986, pp. 569-574.

Mitchell, M P. Bauwens, L. 1990. *Validation of Numerical Models: Empiricism vs. the Laws of Physics*. In: Proc. of the 25th Intersociety Energy Conversion Engineering Conference, Reno Nevada, American Institute of Chemical Engineers, 1990, pp. 424-429.

NASA Glenn Research Center. 2005. *NASA Glenn Research Center. Thermo-Mechanical Systems Branch. Stirling Technology Branch: Background.* http://www.grc.nasa.gov/WWW/tmsb/stirling/doc/stirling_bckgrd.html. Last visited 21/10-2005.

Numerical Recipes. 1997. *Numerical Recipes in FORTRAN: The Art of Scientific Computing.* Cambridge University Press. Online Edition.

Organ, A J. 1992. *Thermodynamics And Gas Dynamics Of The Stirling Cycle Machine.* Cambridge University Press, ISBN 0-521-41363-X.

Powder Metallurgy. 2005. *News & Views: Höganäs AB 2004 Financial Statement*. Powder Metallurgy, Maney Publishing, Vol. 48, No. 1, pp. 1-4

Prieto, J I. Garcia, D. 2005. *Analysis Of Simulations Obtained By Means Of The PROSA Software For Several Prototypes.* In: Proc. of the 12th International Stirling Engine Conference, Durham England, Durham University, 2005, ISBN 0-9535-5582-8, pp. 398-415.

Rauch J S. 1980. *Harmonic Analysis of Stirling Engine Thermodynamics*. In: Proc. of the 15th Intersociety Energy Conversion Engineering Conference, Seattle Washington, The American Institute of Aeronautics and Astronautics, 19801696-1700, pp. 702-707.

Reynolds, W C. 1979. *Thermodynamic Properties in SI*. Department of Mechanical Engineering, Stanford University, ISBN 0-917606-05-1.

Rispoli, F. 1985. *The Lambda-Scheme Method Applied To Stirling Engines.* In: Proc. of the 20th Intersociety Energy Conversion Engineering Conference, Miami Beach, Society of Automotive Engineers, 1985, pp. 3301-3306.

Skelboe, S. 1977. *Extrapolation Methods For Computation Of The Periodic Steady-State Response Of Non-linear Circuits.* IEEE International Symposium on Circuits and Systems Proceedings, 1977, pp. 64-67.

Skelboe, S. 1980. *Computation of the Periodic Steady-State Response of Non-linear Networks by Extrapolation Methods.* IEEE Transactions on Circuits and Systems, vol. CAS-27 No. 3, March 1980, pp. 161-185

Sod, G A. 1978. *A survey of several finite difference methods for systems of non-linear hyperbolic conservation laws.* Journal of Computational Physics (27), Academic Press, pp. 1-31.

Sod, G A. 1985. *Numerical methods in fluid dynamics: Initial and Initial Boundary-Value Problems*. Cambridge University Press, ISBN 0-521-25924-X.

Suratanakavikul, V. Marquis, A J. 1999. *A Comparative Study of Flux-Limiters in Unsteady and Steady Flows.* In: Proc. of the 13th National Mechanical Engineering Conference, South Pattaya, Thailand, 2-3 December 1999.

Tew, R C. Dyson, R W. Wilson, S D. Demko, R. 2004. *Overview 2004 of NASA-Stirling Convertor CFD Model Development and Regenerator R&D Efforts.* NASA Technical report NASA/TM—2004-213404, 2004

Thomas, B. 2000. *Update on the evaluation of different correlations for the flow friction factor and heat transfer of Stirling engine regenerators*. In: Proc. of the 35th International Stirling Engine Conference, American Institute of Aeronautics and Astronautics, Las Vegas, 2000

Thomas, B. 2001. *PROSA - software for evaluation of Stirling cycle machines*. In: Proc. of the 10th International Stirling Engine Conference, Osnabrück Germany, 2001, pp. 67-74.

Thomas, B. 2003. *Calibration routine for Stirling engine simulation program PROSA and comparison of results to experimental data.*In: Proc. of the 11th International Stirling Engine Conference, Rome, University of Rome "La Sapienza", 2001, pp. 29-37.

Thomas, B. 2006. Web page of Prof. Dr.-Ing. Bernd Thomas at the Reutlingen University. http://userserv.fh-reutlingen.de/~thomas/ , visited on 20/3-2006.

Thomsen, P G. 2002. *A Generalised Runge Kutta Method of Order Three.* IMM Technical Report, Department of Mathematical Modelling, Technical University of Denmark.

Toropov, V. 2006. *Development of MARS - Multipoint Approximation method based on the Response Surface fitting*. Web page at http://www.brad.ac.uk/staff/vtoropov/burgeon/b_mars.htm, page linked from the home page of Prof. Vassili Toropov at the University of Bradford (http://www.brad.ac.uk/staff/vtoropov/)

Twinbird. 2006. Free Piston Stirling Cooler homepage of the Twinbird company, http://fpsc.twinbird.jp/en/home_e.html, visited on 13/1-2006.

University of Florida, 2005. *COLAMD web page at the Department of Computer and Information Science and Engineering at University of Florida.* http://www.cise.ufl.edu/research/sparse/colamd/ , visited on 21/11-2005.

Urieli, I. 1983. *A Current Review Of Stirling Cycle Machine Analysis Methods*. In: Proc. of the 18th Intersociety Energy Conversion Engineering Conference, Orlando Florida, American Institute of Chemical Engineers, 1983, pp. 702-707.

Walker, G. Weiss, M H. Fauvel, R. Reader, G. 1990. *Adventures with MarWeiss: A Summary Of Experience With Stirling Simulation.* In: Proc. of the 25th Intersociety Energy Conversion Engineering Conference, Reno Nevada, American Institute of Chemical Engineers, 1990, pp. 342-345.

Wilkinson, B. Allen, M. 1999. *Parallel Programming : Techniques and Applications Using Networked Workstations and Parallel Computers.* Prentice Hall. ISBN 0-13-671710-1.

(This page intentionally left blank)

Paper A:

Preliminary Results from Simulations of Temperature Fluctuations in Stirling Engine Regenerator Matrices

First published at the ECOS 2003 conference:
Andersen, S K. Carlsen, H. Thomsen, P G. Preliminary Results from Simulations of Temperature Fluctuations in Stirling Engine Regenerator Matrices. In: Proceedings of: The 16th International Conference on Efficiency, Cost, Optimisation, Simulation, and Environmental Impact of Energy Systems, Copenhagen, June 30 - July 2, 2003, ISBN: 87-7475-297-9, pp. 1675-1682.

This revised version was accepted on 29/5-2005 for publication in an ECOS 2003 special issue of Energy (Elsevier).

(This page intentionally left blank)

Available online at www.sciencedirect.com
SCIENCE DIRECT®
Energy 31 (2006) 1371–1383
ENERGY
www.elsevier.com/locate/energy

Preliminary results from simulations of temperature oscillations in Stirling engine regenerator matrices

Stig Kildegård Andersen[a,*], Henrik Carlsen[a], Per Grove Thomsen[b]

[a]*Department of Mechanical Engineering, Technical University of Denmark, DK-2800 Lyngby, Denmark*
[b]*Informatics and Mathematical Modelling, Technical University of Denmark, DK-2800 Lyngby, Denmark*

Abstract

The objective of this study has been to create a Stirling engine model for studying the effects of regenerator matrix temperature oscillations on Stirling engine performance.

A one-dimensional model with axial discretisation of engine components has been formulated using the control volume method. The model contains a system of ordinary differential equations (ODEs) derived from mass and energy balances for gas filled control volumes and energy balances for regenerator matrix control masses. Interpolation methods with filtering properties are used for state variables at control volume interfaces to reduce numerical diffusion and/or non-physical oscillations. Loss mechanisms are included directly in the governing equations as terms in the mass and energy balances.

Steady state periodic solutions that satisfy cyclic boundary conditions and integral conditions are calculated using a custom built shooting method.

It has been found possible to accurately solve the stiff ODE system that describes the coupled thermodynamics of the gas and the regenerator matrix and to reliably find periodic steady state solutions to the model. Preliminary results indicate that the regenerator matrix temperature oscillations do have significant impact on the regenerator loss, the cycle power output, and the cycle efficiency and thus deserve further study.

1. Introduction

Computer simulation has been used for design and optimisation of Stirling engines for several decades. Early modelling efforts resulted in very simple models that could be solved on the computers of

* Corresponding author. Tel.: +45 4525 4163.
E-mail address: ska@mek.dtu.dk (S.K. Andersen).

0360-5442/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.energy.2005.05.008

Nomenclature

A	wetted surface area [m^2].
c_p	spec. heat for const. pressure [J/(kg K)]
c_v	spec. heat for const. volume [J/(kg K)]
E	energy [J]
h	conv. heat transfer coeff. [W/(m^2 K)]
m	mass [kg]
$\dot{m}$	mass flow [kg/s]
p	pressure [Pa]
t	time [s]
T	temperature [K]
V	volume [m^3]
cv	in control volume
lbnd	at left boundary of control volume
rbnd	at right boundary of control volume

the day. As computers grow ever more powerful the many assumptions in early models can be gradually relaxed in order to achieve a more accurate prediction of engine behaviour. With more accurate models new insight into phenomena occurring in engines can be gained and engine designs can be optimised to achieve higher thermal efficiencies.

One classical assumption in Stirling engine models is that the temperature of all metal in the engine remains constant during the cycle. In one of the key components in a Stirling engine, the regenerator, the validity of this assumption can be questionable. Analytical studies, such as the studies by Jones [1] suggest that metal temperature oscillations in regenerators can have an impact on engine performance.

Experimental studies of regenerators have mostly been conducted in order to generate correlations for heat transfer and flow friction, such as those compared by Thomas and Pittman [2], that can be used in simulation programs. Several experimental studies, for instance the work of Gedeon and Wood [3] and of Isshiki et al. [4], have been performed by placing small samples of regenerator matrix material into specialised regenerator test equipment instead of performing the measurements on complete regenerators inside actual engines. These studies have yielded information about the flow friction and heat transfer in regenerator matrices but they do not show the influence of regenerator performance on machine performance. In the heat transfer study by Siegel [5], measurements have been performed on a full regenerator inside an actual Stirling engine. But it has only been possible to measure bulk temperatures and pressure losses that result from the combined effects of all phenomena, such as heat transfer, compression/expansion, matrix temperature oscillations, and flow friction, occurring in the regenerator in the engine; The influence of the individual phenomena has not been revealed. Some authors report applying correction terms to simulation results calculated with constant regenerator matrix temperatures in order to account for the effects of regenerator matrix temperature oscillations, see for instance Jones or Kühl and Schultz [1,6].

This paper presents an overview of the method and the initial findings of an effort to explore the effects of dynamic temperature oscillations in regenerator matrices using a numerical simulation model.

2. The stirling engine

Similarly to an internal combustion engine the Stirling engine is based on a gas cycle, where work is expended to compress a cold working gas and where work is extracted by expanding the gas after it has been heated in order to increase the pressure. The Stirling engine, however, works on a closed gas cycle, where the working gas does not take part in combustion and is not exchanged in every cycle. Instead the heating and cooling of the working gas is achieved by sending the working gas back and forth through a serial connection of three heat exchangers, viz. a heater, a cooler, and a regenerator, configured as sketched in Fig. 1.

The efficiency of a Stirling engine can be improved by the regenerator because it can recycle some of the heat that is removed from the gas during transfer to the cold cylinder to preheat the gas when it is transferred back to the hot cylinder. The regenerator is usually made as a porous matrix with a large heat transfer area and a large specific heat capacity from, for instance, metal felt. The regenerator acts as a thermal heat storage that absorbs heat when hot gas flows through it and then releases it again when the flow direction is reversed and the gas flow becomes cooler than the matrix. At steady state periodic operating conditions there is a steep temperature gradient through the regenerator matrix dictated to some degree by the temperatures of the heater and cooler. In order to achieve a high thermal efficiency, it is very important to have a well performing regenerator so that the energy flux loss through the regenerator from the heater to the cooler is minimised. At the same time the pressure drop across the regenerator must be kept as low as possible in order to avoid wasting too much work on pushing the gas back and forth. Regenerator optimisation is thus at the heart of Stirling engine design.

Many practical Stirling engine designs look very different from Fig. 1 but the net effects of compression, expansion, and pushing gas back and forth through heat exchangers and a regenerator are the same.

The engine design parameters and operating conditions that are needed as input parameters for the Stirling engine model used in this study are based on an existing 9 kW engine design by Carlsen and Bovin [7]. A drawing of the region around the cylinder of this engine can be seen in Fig. 2. This engine uses one working piston and a displacer piston instead of the two working pistons in the engine of Fig. 1.

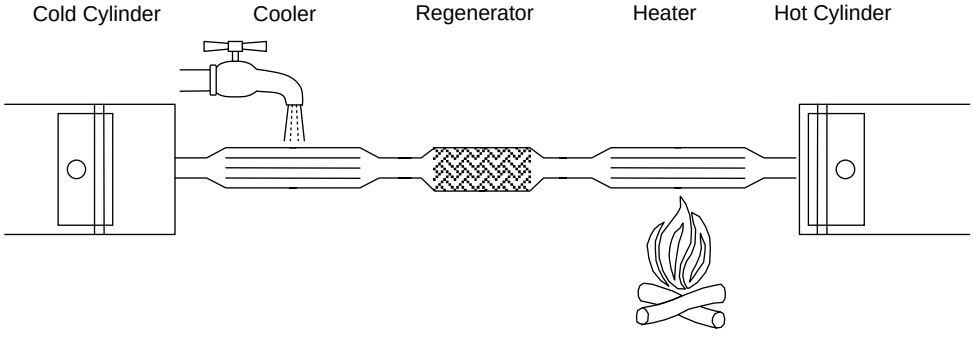


Fig. 1. Stirling engine with two heat exchangers and a regenerator.

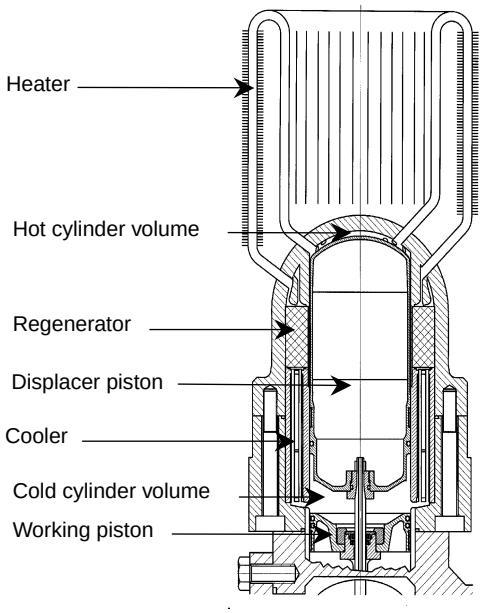


Fig. 2. Cylinder region of 9 kW Stirling engine.

3. The stirling engine model

The Stirling engine model has been formulated as a system of ODEs using the control volume method and conservation of mass and energy.

3.1. Discretisation in model

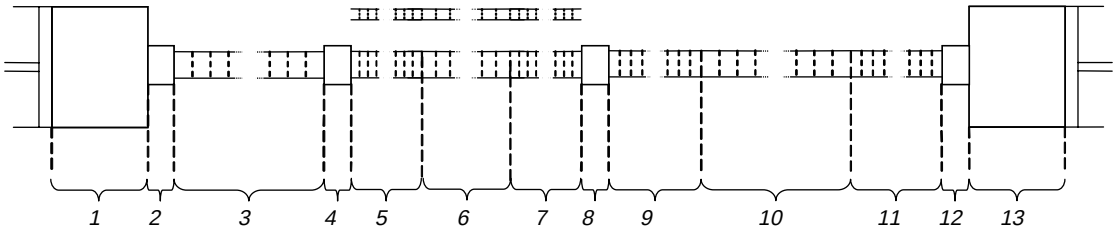
For the sake of simplicity the geometry of the model is not referred to Fig. 2 but to the equivalent geometry in Fig. 1.

The cylinder volumes and the manifold volumes between the components are represented by single control volumes. The sizes of the control volumes representing the cylinder volumes vary in a cyclic fashion.

In the model, the regenerator and the heater are split into three sub components each. In the case of the heater this division has been chosen because the end sections of the heater tubes do not receive the same external heat input as the central part of the tubes. Treating the inactive parts of the heater as separate sub components simplifies the model code. The regenerator has been split into three components to facilitate non-uniform spatial discretisation where finer grids are used in the ends of the regenerator than in the central part. This helps to better resolve localised phenomena in the ends of the regenerator with minimal computational efforts.

The cooler and the sub components in the regenerator and heater are further subdivided into parameterised numbers of control volumes.

The control volumes in the discretisation described above are linked together into a string of control volumes that represents the space in the engine occupied by the working gas.



Components:

1: Cold cylinder volume	3: Cooler	5, 6, 7: Regenerator
13: Hot cylinder volume	9, 10, 11: Heater	2, 4, 8, 12: Manifold volumes

Fig. 3. Discretisation used in Stirling engine model.

A string of control masses that represents the regenerator matrix has also been defined. This string of control masses only spans the length of the regenerator and uses the same axial discretisation as the corresponding control volumes for the working gas.

The resulting discretisation of the engine is illustrated in Fig. 3.

3.2. Phenomena included in model

The working gas is helium and is modelled as an ideal gas with non-constant thermophysical properties.

The model has been formulated using control volumes, control masses, and conservation of mass and energy. Conservation of momentum has been reduced by assuming that the axial inertia of the gas is negligible. More accurately, the velocities of gas flows between control volumes are found using steady state correlations for pressure losses in tube flow, through regenerator matrices, and through flow constraints.

Heat transfer is included between the gas and the wetted metal walls of all components. The instantaneous rates of heat exchange between gas and walls is calculated from the wall temperatures and the average temperatures in the gas filled control volumes using empirical correlations for heat exchange in tube flow, flow through porous matrices and flow in cylinder volumes. The temperatures of the walls in the cylinders, manifold volumes, the heater, and the cooler are assumed fixed at their cycle mean values.

In the regenerator heat transfer is included between the gas and the regenerator matrix. The temperatures of the regenerator matrix control masses can be treated in two different ways. Either the temperatures can be assumed fixed at their cycle mean values or they can be modelled using ODEs derived from an energy balance for a lumped control mass, i.e. a control mass with a uniform temperature. The validity of using a lumped formulation for the matrix thread material has been verified using a separate model that resolves the radial temperature variations inside a thread subjected to the conditions in the regenerator.

Heat conduction inside the metal walls of the engine is also included. This heat conduction affects the temperatures of the metal walls in contact with the gas and thus also the gas. The main heat conduction paths in the model are illustrated by the arrows in Fig. 4.

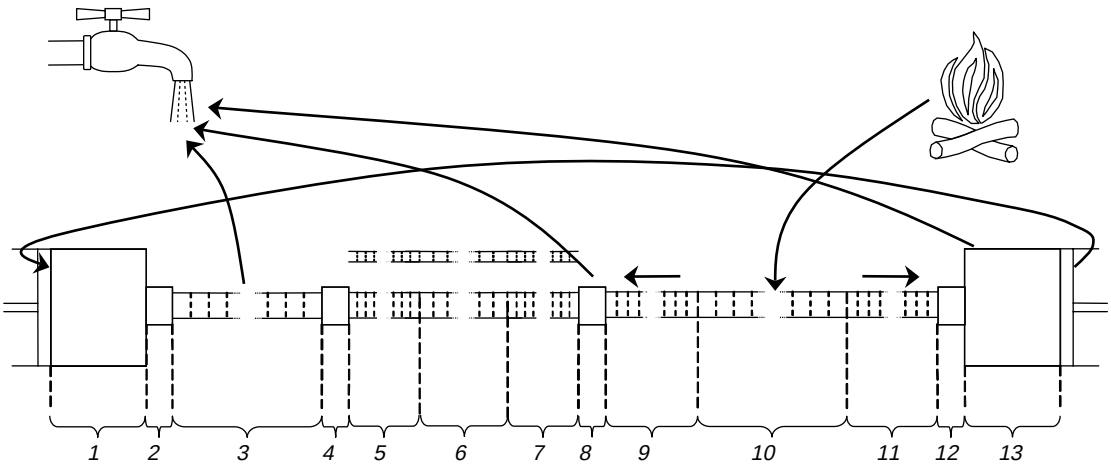


Fig. 4. Heat conduction paths in Stirling engine model.

3.3. Model equation system

The control volumes that represent the working gas are described by two ODEs each, viz. a mass balance and an energy balance. The mass balance ODEs take the form of Eq. (1).

$$\frac{\mathrm{d}m_{\mathrm{cv}}}{\mathrm{d}t} = \dot{m}_{\mathrm{lbnd}} - \dot{m}_{\mathrm{rbnd}} \tag{1}$$

Flows across boundaries are represented as positive in the direction from left to right (from cold to hot) in Fig. 3. The influence of flow friction on the mass distribution in the engine is directly included in the mass balance equations because the mass flow rates are calculated from correlations for pressure losses.

In the energy balances for the control volumes, it has been assumed that kinetic and gravitational potential energies in the control volumes are of negligible magnitudes. The total energy in the control volumes has thus been assumed equal to the internal energy. Because of this the energy balances for the control volumes take the form of Eq. (2).

$$\frac{\mathrm{d}E_{\mathrm{cv}}}{\mathrm{d}t} = \frac{\mathrm{d}(m_{\mathrm{cv}}c_{\mathrm{v}}T_{\mathrm{cv}})}{\mathrm{d}t} = \dot{m}_{\mathrm{lbnd}}c_{\mathrm{p}}T_{\mathrm{lbnd}} - \dot{m}_{\mathrm{rbnd}}c_{\mathrm{p}}T_{\mathrm{rbnd}} + hA_{\mathrm{ht}}(T_{\mathrm{w}} - T_{\mathrm{cv}}) - p_{\mathrm{cv}}\frac{\mathrm{d}V_{\mathrm{cv}}}{\mathrm{d}t} \tag{2}$$

The first two terms in Eq. (2) represent enthalpy flows between neighbouring control volumes. Notice that interpolated boundary temperatures are used in these terms to avoid numerical diffusion. The interpolation is carried out using interpolation methods with filtering properties similar to the mixed interpolation and extrapolation method presented in by Kühl and Schultz [8]. The third term represents convective heat exchange between gas and metal. The last term represents work from volume changes and is only relevant for the cylinder volumes. Energy loss terms such as metal heat conduction typically affect the wall temperatures in the engine and are thus included in the ODEs through the convective heat exchange term.

The control masses that represent the regenerator matrix are described either by ODEs of the form Eq. (2) with only a heat exchange term or by a constant temperature. Heat conduction through the metal in the matrix is not taken into account.

In order to be able to find the temperatures of metal walls throughout the engine it is also necessary to integrate the heat transfer to the wall sections during each cycle. ODEs that have just the heat exchange term of Eq. (2) are used for this purpose.

The governing equations in the model make up a coupled system of first order ODEs.

3.4. Mathematical problem definition

For the purposes of this investigation steady state periodic solutions to the ODE system of the model are needed. In a steady state periodic solution the values of all variables representing masses and energies must be the same at the end and the beginning of the cycle. This requirement is a cyclic boundary condition. The initial masses and energies in the control volumes and control masses must be found so that cyclic boundary conditions are satisfied.

Cyclic heat transfer is assumed to take place for the metal walls in the cylinders, manifold volumes, and in the end sections of the heater. When constant regenerator matrix temperatures are assumed then conditions for cyclic heat transfer are also used for finding the matrix temperatures.

Integrations of heat transfer to or from the wall sections and portions of the regenerator matrix, where cyclic heat transfer is assumed must result in zero net heat transfer if the prescribed constant temperatures are to be true steady state cycle mean values. The fixed wall and matrix temperatures must be found so that this is achieved.

An infinite number of solutions satisfy the above conditions for steady state solutions for the model investigated here. In order to get solutions that are directly comparable, it is necessary to add an additional condition that fixes the total mass of the working gas in the engine. For this purpose, an integral condition that measures the deviation from a user specified mean pressure in the hot cylinder has been chosen. The amount of mass and energy in the string of control volumes representing the working gas must be scaled to eliminate this deviation.

The mathematical problem that must be solved thus contains a cyclic boundary value problem for a system of ODEs, a set of integral conditions for cyclic heat transfer, and an integral condition that must be used for scaling the masses and energies in the gas filled control volumes. The numerical method must also allow additional integrations that can be used for integrating the heat absorbed in the heater and/or cooler and the work done on the pistons, so that performance characteristics, such as power output and thermal efficiency, can be calculated.

4. Implementation of model and numerical method

The mathematical problem of the model is solved numerically using the MusSim (Multi purpose software for Simulation) software by Andersen [9,10]. The software and the model is implemented in standard Fortran95 and has been tested to be compatible with Compaq, Intel, Lahey/Fujitsu, HP, IBM, and Sun compilers and thus runs on a large number of platforms.

A good Stirling engine model should approach periodic steady state asymptotically if integrated forward in time from a given set of initial values, but the convergence may be slow and asymptotical. To drastically accelerate this convergence the MusSim software uses a purpose built shooting method. This shooting method treats the ODEs in the model as a boundary value problem that is solved for initial values and parameters corresponding to a periodic steady state solution to the initial value problem (IVP)

defined by the ODEs. The cyclic boundary conditions and the integral conditions are treated as residual equations that must be solved in much the same way as a Newton–Rhapson method solves non-linear residual equations. The main difference is that the shooting method requires some of the variables to be scaled based on a residual.

The system of ODEs turns out to be very stiff. This is not surprising as the variables in the system have very different inertial properties. For instance, the mass of gas and the energy in a control volume where gas is blowing through at high velocity can change on a much smaller time scale than the thermal energy stored in a control mass in the regenerator matrix. In order to handle this stiffness a suitable IVP method must be used in the shooting method. During this investigation the semi-implicit GERK method by Thomsen [11] has been successfully applied.

To be able to achieve accurate results it has proven necessary to solve ill conditioned non-linear equation systems with very strict tolerances in the internal iterations of the GERK method. This is handled using the built in non-linear equation solver of the MusSim software. It can be time consuming to robustly seek out difficult solutions of this kind but the MusSim software has features to speed up this process and this makes the model manageable for many purposes on a standard PC.

5. Model verification

A great deal of effort has been put into verifying the correctness of the model.

One of the most obvious and also most important tests is to make sure that the model does not leak mass and/or energy due to misconnected control volumes or control masses. Testing for conservation of mass is trivial and has been done by monitoring the sum of the masses of gas in the control volumes. Testing for conservation of energy in the same manner is slightly more involved as it requires the system to be kept adiabatic from the surroundings and it requires the work terms from the movement of the pistons to be zeroed out. For steady state solutions, though, conservation of energy can be verified by an overall energy balance applied at the outer boundaries of the model. These tests are integrated into the model and are performed for every calculated periodic steady state solution. Conservation of mass and energy has been found to satisfy relative tolerances that are stricter than the tolerance enforced by the GERK method for the individual masses and energies.

The correctness of the implementations of the correlations needed in the model have been tested by exchanging them with different correlations and checking the differences caused by the exchanges. The variations have been found to be within the expected ranges.

The calculated performance of the engine according to the assumed best model has been compared to values measured on the actual 9 kW engine. Depending on the combination of correlations used to describe pressure losses and heat transfer the model overestimates the work output from the gas cycle by 8–21% and the efficiency of the gas cycle by 6–18%. These results were expected as the model does not account for certain known losses associated with the displacer piston that divides the cylinder volume of the engine.

6. Preliminary results and discussion

After initial verification of the model had been carried out it was studied how the spatial discretisation in the regenerator influences the calculated temperature profile for the regenerator matrix. During this

investigation the cooler was divided into 10 control volumes and the total length of the heater was divided into 20 control volumes.

For fixed matrix temperatures the total length of the regenerator was divided into 10, 20, and 50 control volumes of uniform size in subsequent simulations. The resulting matrix temperature profiles are shown in Fig. 5. The slightly curved shape of the calculated temperature profiles is largely independent of the fineness of the discretisation.

This experiment was repeated with dynamic regenerator temperatures. In order to present the information from these simulations it has been chosen to plot the maximum and minimum energy based temperatures in the regenerator matrix control masses during a cycle. These plots are shown in Fig. 6 and it is apparent that the shapes of the profiles depend on the fineness of the discretisation. It also appears that the curvatures of the temperature profiles are largest in the ends of the regenerator. This large curvature in the ends of the matrix is caused by the matrix exchanging considerable amounts of energy with incoming gas flows whose temperatures differ significantly from the temperatures of the ends of the matrix.

Based on these observations a new non-uniform discretisation with a larger density of control volumes in the ends of the regenerator was devised. In this discretisation the ends of the regenerator (components 5 and 7 in Fig. 3) were both given lengths corresponding to 8% of the total length of the regenerator while component 6 covered the remaining central part. Component 5 was divided into 10 control volumes, component 6 into five control volumes, and the fineness of the discretisation in component 7, located where the largest gradients were observed in Fig. 6, was varied between 10 and 20 control volumes. Temperature profiles showing the maximum and minimum temperatures in the matrix control masses during the cycle when using the new discretisation are shown in Fig. 7. The temperature profiles in Fig. 7 appear very similar and because of this the discretisation has not been further refined.

Fig. 8 shows comparisons of the temperature profiles for fixed and dynamic matrix temperatures calculated using fine discretisations. The figure shows that the temperature profile in the regenerator matrix looks significantly different when the oscillations of the matrix temperatures are taken into account. The main differences are the temperature oscillations in the ends of matrix and a slightly flatter temperature gradient in the central part of the regenerator.

In order to verify that the solutions with dynamic matrix temperatures found by the shooting method of the MusSim software are true steady state solutions a test procedure was devised.

In this test procedure the initial values found by the shooting method are used as initial values for sequential simulations. These simulations run the engine for an additional 100 consecutive cycles from

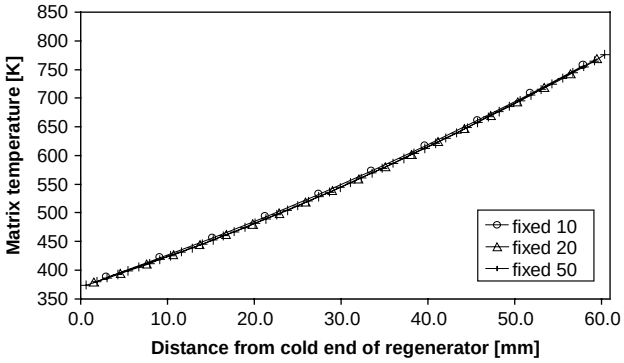


Fig. 5. Axial temperature profiles in the regenerator matrix calculated with fixed matrix temperatures.

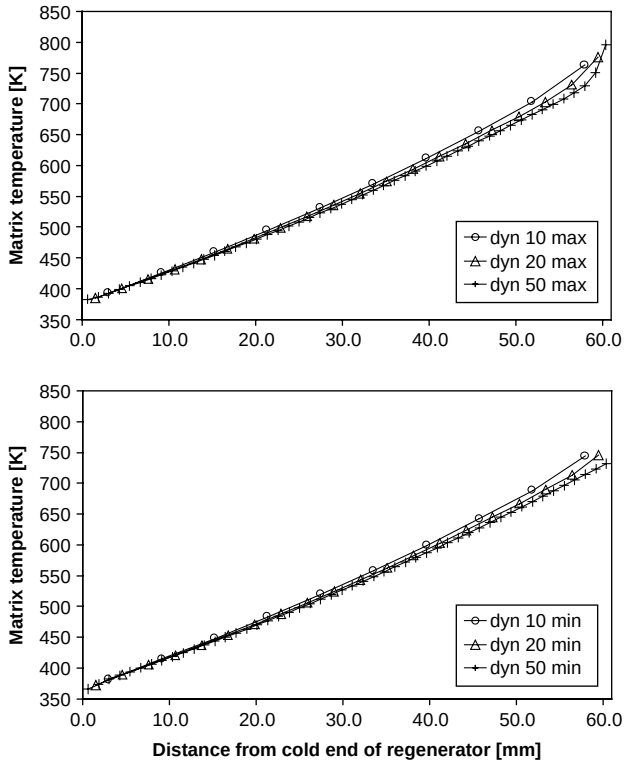


Fig. 6. Max. and min. matrix temperature profiles calculated with dynamic matrix temperatures and uniform discretisation.

the initial values found by the shooting method. After completing the 100 cycles the final values are recorded. The initial values found by the shooting method are then subtracted from these final values and the resulting differences are divided by the initial values. The results of this procedure we label as the relative drifts in the periodic steady state solution.

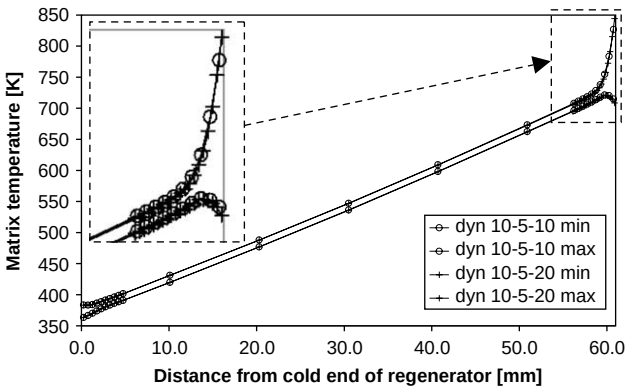


Fig. 7. Max. and min. temperature profiles in the regenerator matrix calculated with dynamic matrix temperatures and non-uniform discretisation.

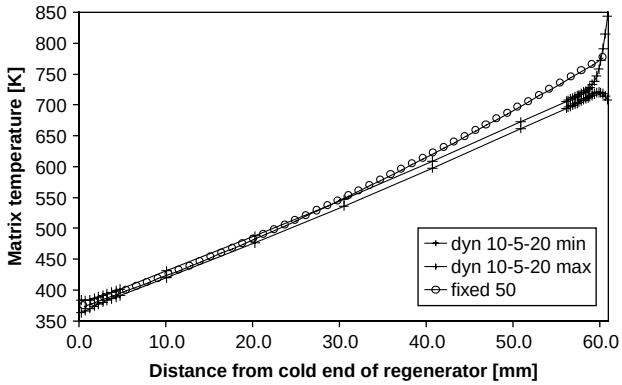


Fig. 8. Comparisons of temperature profiles for fixed and dynamic matrix temperatures.

This test procedure has been completed for the above mentioned five discretisations used in the simulations with dynamic regenerator temperatures. The calculated relative drifts for all variables are shown in Fig. 9 for the case, where shooting has been carried out to a tolerance of 10^{-5} using a relative tolerance of 10^{-6} in the GERK method. The figure shows that the relative drifts of all variables in all the tested solutions are below the tolerance enforced by the shooting method. The figure also shows that the signs of the largest drifts seem random in nature within components. They can thus be interpreted as noise from the integration of the ODEs. The variables most affected by this noise are the masses of gas in the regenerator control volumes and, to a lesser extent, the masses and energies in and close to the cylinder volumes.

The visual inspection of the temperature profiles presented above illustrates that the temperature profiles appear to be shaped differently when oscillations in matrix temperatures are taken into account. The visual inspection does not, however, show that the temperature oscillations have an impact on the performance of Stirling engines. The effect on performance by the temperature oscillations is shown by Table 1 that contains characteristic numbers for the performance of the gas cycle in the engine for all the solutions presented above. The values in the table are the work output, the heat intake from the heater, the thermal efficiency, and the regenerator loss defined as the average net energy flux through the regenerator from the hot end towards the cold end of the regenerator.

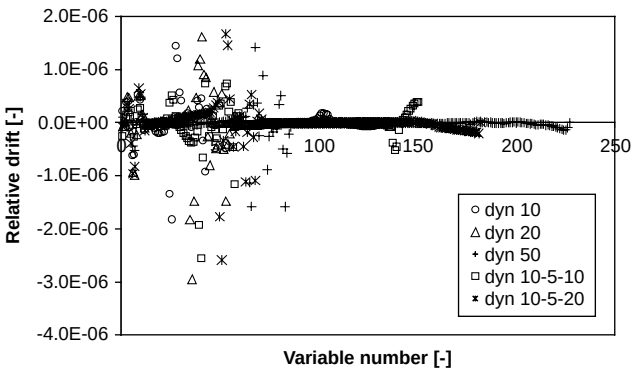


Fig. 9. Relative drifts in periodic steady state solutions.

Table 1
Calculated performance of gas cycle in engine

	Work (kW)	Heat input (kW)	Efficiency (%)	Reg. loss (kW)
Fixed 10	15.213	31.938	47.633	1.519
Fixed 20	15.192	31.886	47.647	1.519
Fixed 50	15.187	31.873	47.650	1.518
Dyn 10	14.617	30.738	47.554	0.543
Dyn 20	14.542	30.622	47.489	0.528
Dyn 50	14.392	30.478	47.222	0.496
Dyn 10-5-10	14.306	30.474	46.946	0.470
Dyn 10-5-20	14.285	30.456	46.904	0.467

The numbers in Table 1 show several distinct differences in performance caused by taking dynamic temperature oscillations in the regenerator matrix into account.

The net energy flux loss through the regenerator is reduced by roughly two thirds. In spite of the reduction of this loss term the thermal efficiency is reduced due to a significant reduction in the power output of the cycle. These differences will be further investigated.

When comparing the above performance numbers for the gas cycle of the engine to the performance of this and other engines it is important to remember that the above performance numbers are for the gas cycle of the engine and not for the shaft power from the engine or for the electrical power output from an engine-generator assembly.

7. Conclusion

It has been found that true steady state periodic solutions can be reliably calculated for a cyclic boundary value problem that describes a Stirling engine and includes the coupled thermodynamics of a gas and a regenerator matrix.

It has also been found that the calculated temperature profile in a regenerator matrix can look significantly different when matrix temperature oscillations are taken into account. The main differences have been found to be significant matrix temperature oscillations in the ends of the regenerator and a less steep matrix temperature gradient in the central part of the regenerator.

Finally, it has been found that the oscillations in the regenerator matrix temperatures influence the calculated performance of the gas cycle in the engine. The observed effects include a reduction of the regenerator loss, the power output, and the thermal efficiency.

Future work will include improvements to the model and the MusSim software as well as comparisons of obtained simulation results with measurements on a test bench.

References

[1] Jones JD. Performance of a Stirling engine regenerator having finite mass. ASME J Eng for Gas Turbines Power 1986;108:669–73.

[2] Thomas B, Pittman D. Update on the evaluation of different correlations for the flow fiction factor and heat transfer of Stirling engine regenerators. In: 35th Intersociety energy conversion engineering conference, Las Vegas; July 24–28 2000, p. 76–84.
[3] Gedeon D, Wood JG. Oscillating-flow regenerator test rig: hardware and theory with derived correlations for screens and felts. NASA Contractor Report 198442, February 1996.
[4] Isshiki S, Takasaki Y, Ushiyama I. An experimental study on flow resistance of regenerator wire meshes in oscillating flow. In: 32nd Intersociety energy conversion engineering conference, Honolulu; July 27–August 1 1997, p. 1027–32.
[5] Siegel A. Experimental investigations on the heat transfer behaviour of wire mesh regenerators in an oscillating flow. In: Proceedings of the European Stirling Forum, 2000.
[6] Kühl HD, Schulz S. A 2nd order regenerator model including flow dispersion and bypass losses. In: Proceedings of the 31st intersociety energy conversion engineering conference, Washington DC; 11–16 August 1996.
[7] Carlsen H, Bovin J. Test of 9 kW Stirling engine using biogas as fuel. In: Proceedings of the 10th international Stirling engine conference, VDI, Osnabrück, September 2001.
[8] Kühl HD, Schulz S. Modelling of thermal regenerator losses due to axial flow dispersion. In: Proceedings of the 7th international conference on Stirling cycle machines, Tokyo, November 5–8 1995.
[9] Andersen SK. Software for simulation of thermodynamic systems. Preparatory Thesis, Department of Mechanical Engineering, DTU Denmark, 2002.
[10] Andersen SK. Simulation and optimisation of reciprocating engines. Master's Thesis Project, Department of Mechanical Engineering, DTU Denmark, 2002.
[11] Thomsen PG. A generalized Runge Kutta Method of Order Three, Technical Report no. IMM-REP-2002-07. Department of Mathematical Modelling, DTU Denmark, 2002.

(This page intentionally left blank)

Paper B:

Numerical study on optimal Stirling engine regenerator matrix designs taking into account the effects of matrix temperature oscillations

Andersen, S K. Carlsen, H. Thomsen, P G. Numerical study on optimal Stirling engine regenerator matrix designs taking into account the effects of matrix temperature oscillations. Energy Conversion and Management, Vol. 47/7-8, 2006, pp. 894-908.

(This page intentionally left blank)

Available online at www.sciencedirect.com
SCIENCE DIRECT®
Energy Conversion and Management 47 (2006) 894–908
ENERGY CONVERSION & MANAGEMENT
www.elsevier.com/locate/enconman

Numerical study on optimal Stirling engine regenerator matrix designs taking into account the effects of matrix temperature oscillations

Stig Kildegård Andersen [a,*], Henrik Carlsen [a], Per Grove Thomsen [b]

[a] *Department of Mechanical Engineering, Energy Engineering Section, Technical University of Denmark, Nils Koppels Allé bygning 402, DK-2800 Kgs. Lyngby, Denmark*
[b] *Informatics and Mathematical Modelling, Technical University of Denmark, Richard Petersens Plads bygning, DK-2800 Kgs. Lyngby, Denmark*

Received 12 December 2004; accepted 10 June 2005
Available online 1 August 2005

Abstract

A new regenerator matrix design that improves the efficiency of a Stirling engine has been developed in a numerical study of the existing SM5 Stirling engine. A new, detailed, one-dimensional Stirling engine model that delivers results in good agreement with experimental data was used for mapping the performance of the engine, for mapping the effects of regenerator matrix temperature oscillations, and for optimising the regenerator design. The regenerator matrix temperatures were found to oscillate in two modes. The first mode was oscillation of a nearly linear axial matrix temperature profile while the second mode bended the ends of the axial matrix temperature profile when gas flowed into the regenerator with a temperature significantly different from the matrix temperature. The first mode of oscillation improved the efficiency of the engine but the second mode reduced both the work output and efficiency of the engine. A new regenerator with three differently designed matrix sections that amplified the first mode of oscillation and reduced the second improved the efficiency of the engine from the current 32.9 to 33.2% with a 3% decrease in power output. An efficiency of 33.0% was achievable with uniform regenerator matrix properties.

* Corresponding author. Tel.: +45 4525 4122; fax: +45 4593 5215.
E-mail address: ska@mek.dtu.dk (S.K. Andersen).

0196-8904/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.enconman.2005.06.006

Keywords: Stirling engine; Regenerator; Matrix temperature oscillations; Simulation

1. Introduction

The Stirling engine is a closed cycle, regenerative, external combustion engine. In a Stirling engine, work is expended to compress a cold gas, and the gas is then heated to further increase the pressure. The hot, high pressure gas is then expanded, and more work can be extracted than was needed to compress the cold gas. Finally, the gas is cooled before the next cycle begins with a new compression. The heating and cooling of the working gas is achieved by sending the working gas back and forth through a serial connection of heat exchangers, i.e. a cooler, a regenerator and a heater. The regenerator is a void filled with a porous matrix with a large heat transfer area and a large heat capacity, made, for instance, from metal felt. The main purpose of the regenerator is to act as a thermal heat storage that minimises the amount of energy that must be added in the heater, thereby increasing the thermal efficiency. The regenerator does this by absorbing energy when the gas is flowing from the heater towards the cooler and releasing the energy again when the gas is flowing from the cooler towards the heater. Because the matrix alternately absorbs and releases energy, the temperature profile of the matrix oscillates in time.

Because of their large heat transfer area, porous matrix regenerators also, typically, greatly increase the heat transfer in an engine and, hence, increase the power output. Regenerators and their design are, thus, central to the performance of Stirling machines, and their influence on the performance of the machines has been studied intensively. Both efficiency and a high power output are important because they influence the cost of power produced by an engine. Furthermore, the efficiency directly influences the environmental impact per unit produced power. This paper presents a new way to design regenerator matrices so that Stirling engines can achieve higher efficiencies while maintaining a high power output. The new design is based on the results of a numerical study on the influence of the regenerator matrix temperature oscillations on the performance of a Stirling engine.

1.1. Previous regenerator studies

Experimental studies of regenerators have mostly been conducted in order to generate correlations for heat transfer and flow friction, such as those compared by Thomas and Pittman [1], that can be used in simulation programs. Several experimental studies, for instance, the work of Gedeon and Wood [2] and of Isshiki et al. [3], have been performed by placing small samples of regenerator matrix material into specialised regenerator test equipment instead of performing the measurements on complete regenerators inside actual engines. These studies have yielded information about the flow friction and heat transfer in regenerator matrices, but they do not show the influence of regenerator performance on machine performance. In the heat transfer study by Siegel [4], measurements have been performed on a full regenerator inside an actual Stirling engine, but it has only been possible to measure bulk temperatures and pressure losses that result from the combined effects of all phenomena, such as heat transfer, matrix temperature oscillations and flow friction, occurring in the regenerator in the engine; the influence of the individual phenomena has not been revealed.

Analytical and semi-analytical studies of Stirling machines require that the equations and boundary conditions of the models be simple enough that analytical solution methods can be successful applied. Hence, significant simplifications compared to real world conditions are needed, and the agreement between predicted and actual machine performance suffers accordingly. Analytical models, however, are useful for identifying phenomena and for predicting trends. In 1982 and 1986, Jones used an analytical model with a piece wise linear axial matrix temperature profile to show that matrix temperature oscillations influence the performance of Stirling engines because they induce a heat pumping effect and cause a loss of power output [5,6]. More recently, Bauwens has shown that thermoacoustic effects in regenerators can be significant when the flow passages in the regenerator are of significant magnitude compared to the heat penetration depth in the gas [7].

Detailed numerical studies of regenerators, see, for instance, the work of Gary et al. from 1984 [8], have been performed. When the regenerator models are not integrated into detailed and complete machine models, the influence of the regenerators on their own boundary conditions and on machine performance cannot be determined.

1.2. Regenerator matrix design

The design of metal felt regenerators can be characterised by the geometry of the regenerator volume, the diameter of the wire from which the metal felt is made and the fill factor (or, conversely, the void fraction) of the metal felt. In this study, only the wire diameter and fill factor are considered.

The amount of heat that must be added in the heater of a Stirling engine can be reduced by decreasing the net average flux of energy that is carried from the heater to the cooler by the gas in the engine, i.e. by reducing the regenerator loss. The regenerator loss can be reduced by reducing the temperature difference between the gas and the matrix by increasing the heat transfer area per unit volume. The heat transfer area can be increased by reducing the wire diameter and/or by increasing the fill factor but doing so will increase the flow resistance. Pressure losses caused by flow resistance in the regenerator cause a loss of power output from the engine.

When a regenerator design is optimised without taking into account the effects of matrix temperature oscillations on engine performance, a design that balances the heat transfer properties and the flow resistance in the matrix can be found. In this study, matrix temperature oscillations are taken into account and regenerator designs that take into account the thermal inertia of the matrix in addition to the heat transfer properties and flow resistance are discussed.

1.3. The effects of matrix temperature oscillations

The findings of Jones [6] suggest that the magnitudes of the heat pumping and power loss induced by regenerator matrix temperature oscillations are inversely proportional to the heat capacity of the matrix and, hence, to the magnitudes of the matrix temperature oscillations. The study of Jones does not account for the influence of the matrix temperature oscillations on the boundary conditions of the regenerator. We have presented results [9] that indicate that the inverse proportionalities are only approximate. Still, this dependence on the heat capacity of the matrix makes it possible to extrapolate reliably to the case of infinite matrix heat capacity from simulations performed with finite matrix heat capacity. If, for instance, simulations are performed where the heat

capacity of the matrix is increased by a factor of 100, the magnitudes of the effects of matrix temperature oscillations are reduced to the fraction 0.01; the extrapolation from this case to the case of infinite matrix heat capacity is a small one.

1.4. The present study

The effects of regenerator matrix temperature oscillations on the performance of a Stirling engine have been studied by using a detailed Stirling engine model to map the performance of an existing Stirling engine and the effects of matrix temperature oscillations for a wide range of regenerator matrix designs. The information gained about the influence of different modes of matrix temperature oscillations on the performance of the engine leads to a suggestion for a new regenerator matrix design where the regenerator is divided into three sections with different matrix designs. The new regenerator design allows the engine to obtain a higher efficiency than can be achieved with a regenerator with a uniform matrix design while maintaining a high power output.

2. Methods

2.1. The studied Stirling engine

The engine design parameters and operating conditions used as input for the Stirling engine model in this study were based on the 9 kW β-type Stirling engine, SM5 [10]. The engine is a hermetically sealed unit and has a generator built into a sealed and pressurized crank case. The regenerator of the engine is an annular shaped void with a stainless steel felt matrix. A picture of the engine and a drawing of the cylinder region of the engine are shown in Fig. 1.

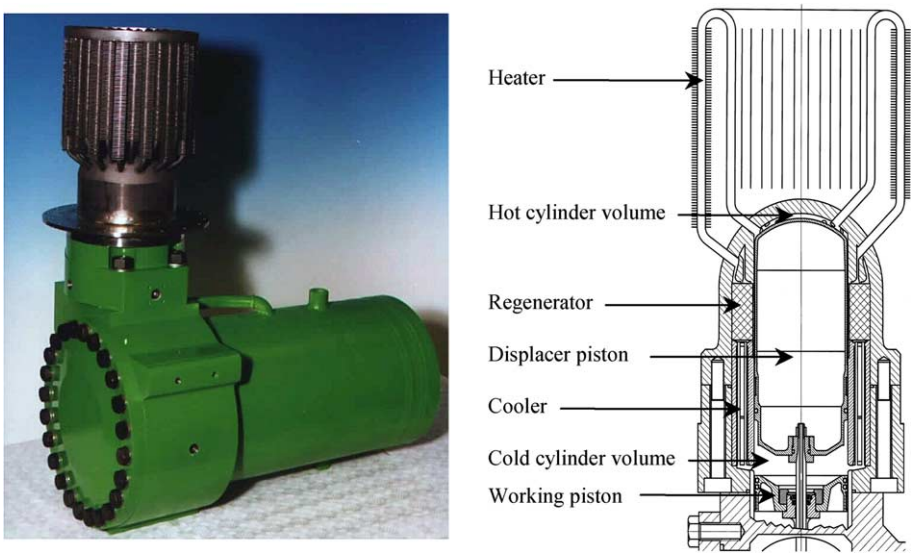


Fig. 1. The SM5 Stirling engine with exposed heater (left) and a drawing of the cylinder region of the engine.

2.2. The Stirling engine model

In this study, we used a new Stirling engine model that we have documented [9] is capable of delivering results that are in good agreement with experimental data over a wide range of operating conditions with both helium and nitrogen as working gas. This Stirling engine model is a refinement of the modelling approach we first presented in Ref. [11]. We have documented [9,11] that the model is able to reproduce the heat pumping and power loss effects caused by the matrix temperature oscillations that were originally predicted by Jones [5,6].

The outer boundary conditions of the model are the temperature profile on the outside of the heater tubes, the power outlet from the generator and the flow rate and temperature of the cooling water. Because the model does not include the burner system, the heat intake and electrical efficiency calculated by the model is based on the heat absorbed by the heater tubes and do not include burner losses.

In the model, an equivalent one-dimensional geometry is used to represent the working volume of the Stirling engine, i.e. the computational domain. The computational domain includes the displacer piston clearance gap and the manifolds of the engine in addition to the cylinder volumes, cooler, regenerator and heater. In the displacer piston clearance gap, the control volumes follow the motion of the displacer piston, but everywhere else in the engine, the control volumes are fixed in space. The discretization of the computational domain into control volumes was locally refined where solutions contain large gradients, and this was done so that grid convergent solutions were obtained throughout the studied range of regenerator designs. The discretization of the computational domain is illustrated in Fig. 2.

In formulating the governing equations for the gas in the computational domain, the time dependent forms of the balance equations for mass, energy and momentum were used for writing ordinary differential equations (ODEs) that describe the gas in the control volumes. The balance equation for momentum was applied on a staggered mesh, i.e. to a second set of control volumes centred on the boundaries of the control volumes used for the mass and energy balances. The ODEs for the mass and energy balances were transformed into ODEs for pressure and temperature using the ideal gas equation of state. ODEs for the flow velocities were then derived from the balance equations for momentum. Asymmetric interpolation methods with filtering properties, similar to the quadratic polynomial method presented by Kühl and Schultz in Ref. [12], were

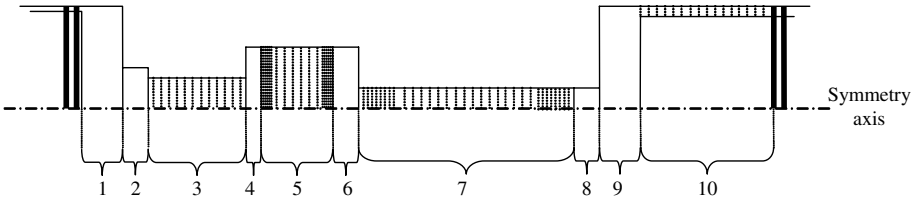


Fig. 2. Discretization of computational domain used in Stirling engine model.

included in the model to minimise numerical diffusion of quantities that are transported by advection. Artificial dissipation based on the 2nd and 4th order spatial derivatives of the flow velocities was included to dissipate acoustic phenomena with short wavelength. We have previously described [13] this method for modelling the gas more closely.

All wetted surfaces of the engine components are divided into control masses that interact with the gas filled control volumes. Most of the metal control masses correspond spatially to the gas filled control volumes, but in the displacer piston clearance gap, the control volumes that resolve the gas in the gap slide over the control masses that resolve the cylinder wall. The temperatures of the control masses containing the regenerator matrix material are modelled as dynamic, and the temperatures of all the remaining control masses are modelled as constant at their periodic steady state mean values. The periodic steady state mean values are found from the integral conditions that the net heat transfer to the control masses during one engine cycle must be zero. The control masses containing the regenerator matrix material are modelled using ODEs derived from an energy balance for a lumped control mass, i.e. a control mass with a uniform temperature. The validity of using a lumped formulation and, hence, of assuming that radial temperature gradients inside the regenerator matrix wires are not significant has been verified using a separate model that resolves the radial temperature variations inside a single matrix wire. In this study, the heat transfer rates between the gas and matrix and the flow friction were calculated using the correlations by Kühl presented by Thomas and Pittman [1].

Losses caused by flow friction and heat losses are coupled directly into the governing equations in the model by including their effects in the balance equations of the model formulation. Flow friction, for instance, is included as terms in the momentum balance equations. Heat conduction in the walls of the engine components, to mention another example, affects the energy balances of the metal control masses in the model and, hence, the temperatures of the control masses. Because the control masses interact with the gas filled control volumes through convective heat exchange, the heat conduction in the walls of the engine components is also coupled into the governing equations for the gas. Coupling loss terms directly into the governing equations breaks with more traditional Stirling engine modelling approaches, such as the approach of Uriele and Berchowitz [14], where losses are assumed to be decoupled from each other and from the governing equations and, hence, can be applied as correction terms to the calculated performance of an idealised engine.

Heat transfer and flow friction are calculated using empirical correlations for heat transfer and flow friction inside tubes, flow constraints, engine cylinders and regenerator matrices. Correlations derived for steady state conditions are used for tube flow. Approximated velocity and temperature profiles are used for calculating friction and heat transfer in the displacer piston clearance gap. The working gas and the steel in the engine are modelled with temperature dependent thermophysical properties. The bearing and seal friction forces are calculated from the forces exerted on the pistons by the gas in the engine. The efficiency of the generator is assumed to be load independent.

2.3. The simulation tool for computing steady state solutions

Periodic steady state solutions to the model were computed by formulating a boundary value problem (BVP) in the governing equations of the model and then applying the shooting method of

the *MusSim* software to solve the BVP. *MusSim*, or *Multi Purpose Software for Simulation*, is a general purpose simulation tool being developed in house at the Department of Mechanical Engineering at the Technical University of Denmark. A paper describing the shooting method of the *MusSim* software has been submitted for publication [15].

2.4. Engine operating conditions

The input to the model defined operating conditions where the engine operated at 1025 rpm with helium as the working gas at a mean pressure of 7.9 MPa. The temperature profile on the outside of the heater tubes spanned between 640 °C in the ends nearest the regenerator and 760 °C near the ends that connect to the hot cylinder volume. The cooling water had an average temperature of 36 °C. These conditions are near optimal operating conditions for the SM5 engine where the engine delivers 10.7 kW of electric power with an efficiency of almost 33%.

2.5. Mapping of engine performance

The studied range of regenerator designs was defined by the range of wire diameters from 15 to 150 μm and the range of fill factors from 0.01 to 0.27 (equivalent to void fractions from 0.99 to 0.73). The mapping of engine performance was done by simulating the SM5 engine on a mesh with 72 regenerator matrix design points defined by the nine wire diameters 15, 20, 35, 50, 65, 100, 150, 200 and 250 μm and the eight fill factors 0.01, 0.02, 0.04, 0.07, 0.12, 0.17, 0.22 and 0.27. The regenerator currently installed in the SM5 engine has a wire diameter of 60 μm and a fill factor of 0.22.

2.6. Mapping of the effects of matrix temperature oscillations

The specific heat of the matrix material was varied by multiplying the temperature dependent specific heat of the stainless steel with a factor. Simulations were performed with this multiplication factor equal to 2, 10 and 100 for each of the 72 regenerator design points. Together with the simulations performed to map the performance of the engine, this yielded a total of four performance data sets for the different matrix heat capacities for extrapolating to the cases of no matrix temperature oscillations. In order to take into account deviations from linear dependence between the performance data and the reciprocal of the matrix heat capacity, the extrapolations were done in the reciprocal of the matrix heat capacity using cubic polynomials through the four data points at each of the regenerator designs. The effects of the matrix temperature oscillations on the performance of the engine were isolated by subtracting the extrapolated results for the cases of no matrix temperature oscillations from the mapping of the performance of the engine.

2.7. Regenerator design optimisation

Two optimisations of the regenerator matrix were performed using the efficiency of the engine as the objective function to be maximised. In the first optimisation, the efficiency was optimised by adjusting the wire diameter and fill factor uniformly throughout the matrix. In the second optimisation, the regenerator matrix was divided into three sections where the two end sections were

each 5 mm long, leaving a central section of 51 mm. The lengths of 5 mm for the end sections were chosen for convenience; the 5 mm end sections correspond to the sections used to refine locally the discretization in the ends of the regenerator. In the second optimisation, the efficiency was optimised by adjusting the individual wire diameters and fill factors for the three sections. The optimisations were performed using a conjugate gradients method available in the *MusSim* software.

3. Results

3.1. Axial temperature profile in the regenerator matrix

Fig. 3 shows the computed axial matrix temperature profile in the regenerator currently installed in the engine at the chosen operating conditions. The matrix temperature profile is shown as the minimum and maximum temperatures reached during the cycle at the centres of the matrix control masses. A curve extrapolated to the case of infinite matrix heat capacity is also shown. Fig. 3 shows that the axial temperature profile was almost linear in the central part of the regenerator, and that the slope of the profile was slightly less steep when matrix temperature oscillations were taken into account. The figure also shows that the matrix temperatures oscillated approximately 12 °C in the central part of the matrix, and that larger temperature oscillations occurred in the ends of the matrix. In the hot end of the regenerator, the matrix temperature oscillations were as large as 70 °C.

3.2. Mapping of engine performance

Fig. 4 shows contour plots of the calculated power output, electrical efficiency, regenerator loss and heat intake of the SM5 Stirling engine for the range of regenerator designs studied. The horizontal axis in the plots represents the fill factor, and the vertical axis represents the wire diameter in the matrix. On horizontal lines in the plots, the ratio of heat transfer area to heat capacity is constant, and on vertical lines in the plots, the heat capacity is constant.

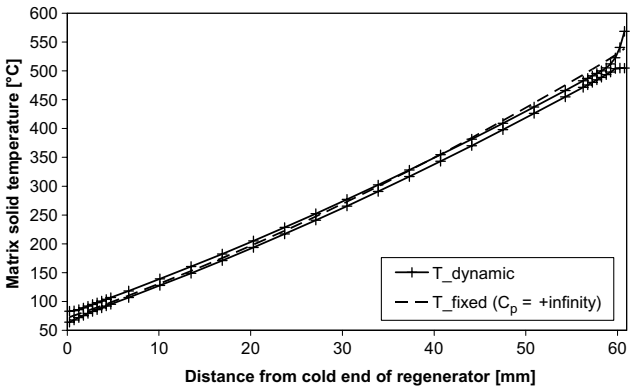


Fig. 3. Minimum and maximum matrix temperatures reached during a cycle in the current regenerator of the SM5 Stirling engine, and the corresponding matrix temperature profile for infinite matrix heat capacity.

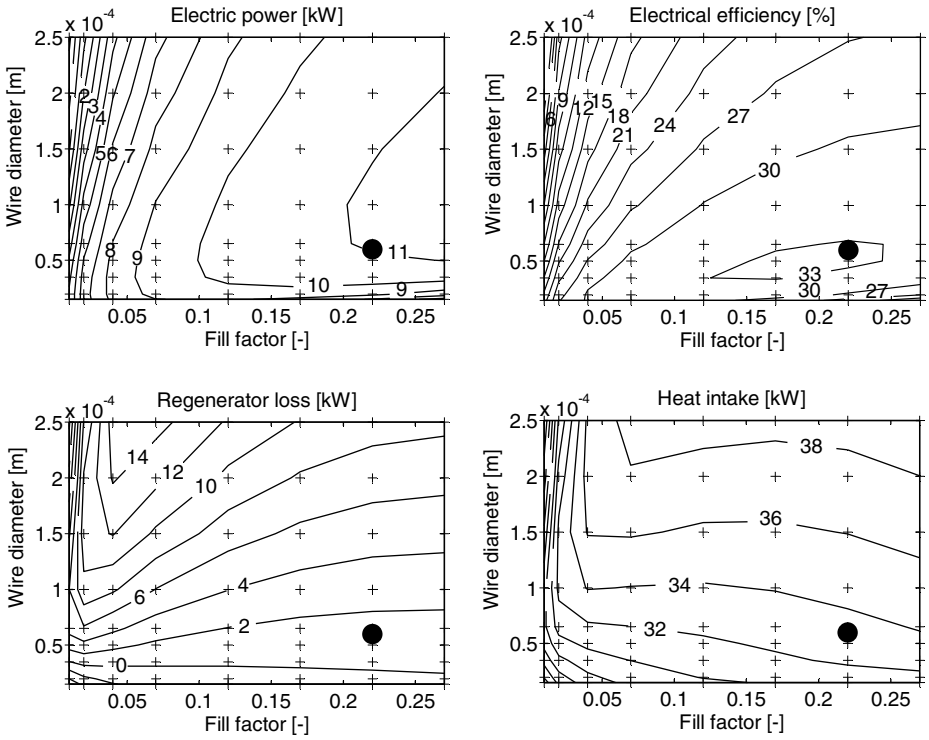


Fig. 4. Contour plots of the power output, electrical efficiency, regenerator loss and heat intake of the SM5 Stirling engine for regenerator designs with wire diameters ranging from 15 μm to 250 μm and fill factors from 0.01 to 0.27. The crosses mark the locations of calculated values. The dots mark the current regenerator design.

Fig. 4 shows that the power output from the engine decreased sharply for low fill factors and large wire diameters where the heat transfer area in the matrix was relatively small. A decrease in the power output also occurred for very small wire diameters and relatively large fill factors where the pressure drop across the regenerator was large. The contour plot for the electrical efficiency shows a peak near the present regenerator design.

For fill factors above approximately 0.07, the regenerator loss increased with smaller fill factors and larger wire diameters, i.e. with smaller heat transfer area in the matrix. For fill factors below 0.05, the regenerator loss decreased with decreasing fill factor for wire diameters above 50 μm. The heat intake of the engine showed similar trends, but for fill factors above 0.05, the variations with wire diameter were smaller in magnitude and the dependence on the fill factor was less pronounced.

3.3. Mapping of the effects of matrix temperature oscillations

Fig. 5 shows contour plots of the influences of the matrix temperature oscillations on the power output, electrical efficiency, regenerator loss and heat intake of the SM5 Stirling engine for the studied range of regenerator designs.

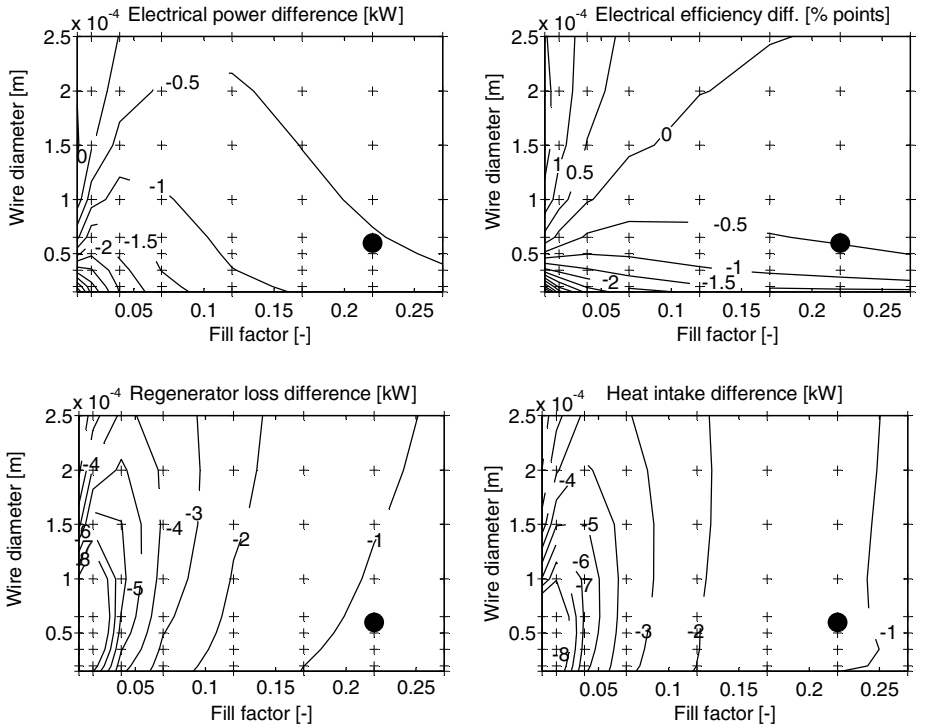


Fig. 5. Contour plots of the differences in power output, electrical efficiency, regenerator loss and heat intake for the SM5 Stirling engine caused by the matrix temperature oscillations for regenerator designs with wire diameters ranging from 15 μm to 250 μm and fill factors from 0.01 to 0.27. The crosses mark the locations of calculated values. The dots mark the current regenerator design.

Fig. 5 shows that the matrix temperature oscillations reduced the electrical power output of the engine for most of the explored regenerator designs, and that the reduction was largest for low fill factors and small wire diameters. In very sparse regenerator matrices with relatively large wire diameters, the electrical power output was slightly increased by the matrix temperature oscillations. At the current regenerator design, the matrix temperature oscillations reduced the electrical power output of the engine by approximately 5%. The electrical efficiency was increased by the matrix temperature oscillations for low fill factors and large wire diameters and reduced for larger fill factors and small wire diameters.

In Fig. 5, the plots for the differences in the regenerator loss and the heat intake look very similar, suggesting a strong coupling between them. The regenerator loss and the heat intake were reduced the most by the matrix temperature oscillations when the fill factor was low. The decrease was largest for small wire diameters.

3.4. Regenerator design optimisation

The optimisation of the regenerator where the design of the matrix was kept uniform throughout the matrix showed that the electrical efficiency could be improved by 0.1% points from 32.9%

to 33.0% by reducing the fill factor from 0.22 to 0.185 and reducing the wire diameter from 60 μm to 49 μm. This design change reduced the power output of the engine by 2% and reduced the regenerator loss by 31% from 1.15 kW to 0.79 kW.

The optimisation where the regenerator matrix was split into three sections resulted in a design with a fill factor of 0.22 and a wire diameter of 90 μm in the cold end section, a fill factor of 0.15 and a wire diameter of 36 μm in the central section and a fill factor of 0.21 and a wire diameter of 84 μm in the section at the hot end of the regenerator. With this regenerator design, the electrical efficiency of the engine was improved by 0.3% points from 32.9% to 33.2%, while the power output was reduced by 3% and the regenerator loss was reduced by 5% from 1.15 kW to 0.52 kW.

4. Discussion

4.1. The matrix temperature profile and temperature oscillations

The matrix temperature oscillations shown in Fig. 3 appear to consist of two contributions: an overall oscillation and additional oscillations near the ends of the regenerator. The overall oscillation did not bend the axial matrix temperature profile but only shifted it up and down. It would be the only contribution if the ratio of heat transfer to heat capacity was constant throughout the regenerator. The additional oscillations near the ends of the regenerator did bend the axial matrix temperature profile. They were induced when the temperature difference between the matrix and the gas flowing into the regenerator was significantly different from the temperature difference between the matrix and the gas in the central part of the regenerator. They are denoted as the inflow induced matrix temperature oscillations in the remainder of this paper. The characteristics of the shape of the temperature profile changed slightly when the fill factor was 0.01 because the cold end of the profile straightened slightly at the gas inflow temperature during inflow from the cooler.

The magnitudes and the penetration depths of the inflow induced matrix temperature oscillations depended on the design of the matrix. When the wire diameter was reduced, the heat transfer area and the ratio of heat transfer area to heat capacity were increased. This reduced the penetration depths of the inflow induced temperature oscillations and increased their magnitudes. When the fill factor was increased, the heat capacity of the matrix was increased, but the ratio of heat transfer area to heat capacity did not change. This reduced the penetration depths of the inflow induced temperature oscillations without having much effect on their magnitudes.

The magnitudes of the overall matrix temperature oscillations depended almost entirely on the heat capacity of the matrix and, therefore, on the fill factor.

4.2. The effects of matrix temperature oscillations

Fig. 5 shows that the largest decrease in power output from the matrix temperature oscillations was found for low fill factors and small wire diameters where both the inflow induced- and overall matrix temperature oscillations were largest. At the same time, the electrical efficiency was increased by the matrix temperature oscillations for low fill factors and large wire diameters and

reduced for larger fill factors and small wire diameters. Hence, the electrical efficiency was increased where the overall temperature oscillations were largest and the inflow induced temperature oscillations were smallest, and vice versa. These observations suggest that the inflow induced matrix temperature oscillations had a negative influence on both power output and electrical efficiency, while the overall matrix temperature oscillations sometimes had a positive influence on electrical efficiency. Since positive effects were found, it was concluded that, of the heat pumping and the power loss due to matrix temperature oscillations documented by Jones in Ref. [6], the first could be predominant in the SM5 engine when the overall matrix temperature oscillations were large.

4.3. A new regenerator design with three sections

The optimisation of a regenerator divided into three sections showed that a notably higher electrical efficiency could be achieved than with a uniform matrix design. The optimisation yielded a matrix in which the fill factors and thread diameters in the end sections of the matrix were larger than in the central section. In this design, the ratio of heat transfer area to heat capacity was smaller in the ends of the matrix than in the central part, and the ratio was also smaller than in the optimal uniform regenerator design. The smaller ratio of heat transfer area to heat capacity resulted in smaller inflow induced matrix temperature oscillations. The central section of the matrix had a smaller wire diameter and a smaller fill factor than the optimal uniform matrix design, and this caused the overall matrix temperature oscillations in the central part of the matrix to be larger. The differences in the magnitudes of the inflow induced- and overall matrix temperature oscillations can be seen in Fig. 6. In addition to balancing heat transfer and pressure losses, the new matrix design with three sections, thus, reduced the negative effects of the inflow induced matrix temperature oscillations and intensified the positive effects of the heat pumping driven by the overall matrix temperature oscillations.

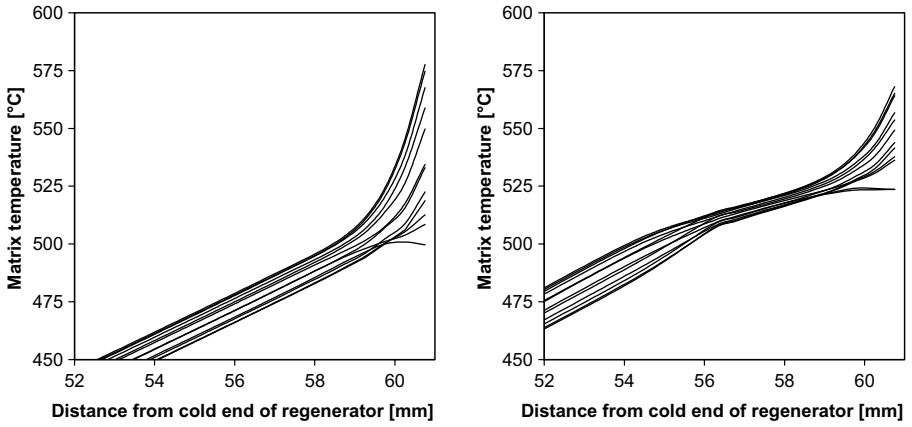


Fig. 6. Axial matrix temperature profiles in hot end of regenerator plotted every 30° crank angle for an optimised matrix with uniform properties (left) and an optimised matrix with three sections (right).

4.4. Engine performance with uniform matrix

The mapping of engine performance in Fig. 4 shows that the current regenerator of the SM5 engine is already well optimised with respect to electrical efficiency. Optimisation showed that only a 0.1% point improvement of electrical efficiency was achievable with uniform wire diameter and fill factor throughout the matrix. Fig. 4 also shows that the work output of the engine could be increased by increasing both the wire diameter and the fill factor compared to the current matrix design and that doing so would have only a moderate impact on the efficiency of the engine.

4.5. Uncertainties and the generality of the results

The largest source of uncertainty of the results was the Stirling engine model itself; the shooting method used for finding periodic steady state solutions to the model delivers accurate solutions to the model as we documented in Ref. [15]. Because the model is one-dimensional, it yields little information about gradients transverse to the main flow direction, and it cannot resolve the flow patterns in open volumes such as cylinder volumes. Heat transfer and flow friction calculations, therefore, depended entirely on empirical correlations and their accuracy for the flow conditions in the model in the simulations. The largest uncertainty in this regard was the use of steady state correlations for heat transfer and flow friction in the tubular heat exchangers. It was also not certain that transverse gradients in the flow channels of the sparsest regenerator matrices studied were properly taken into account. However, the regenerator designs that yielded optimal performance were both relatively close to the present regenerator design, where we have documented that the model accurately predicts the performance of the engine [9].

The deviations from linear dependencies between the performance characteristics for the engine and the reciprocal of the matrix heat capacities were small, and using cubic polynomial extrapolations to the cases of infinite matrix heat capacities was, thus, adequate.

It has been documented using simple axial matrix temperature profiles that heat pumping and power loss effects are caused by matrix temperature oscillations, and the same effects have been observed in models that resolve more intricately shaped matrix temperature profiles. Hence, it appears likely that the effects of, at least, the overall matrix temperature oscillations can be expected to be general and to follow the trends, such as dependence on phase angle between mass flow variation and pressure oscillation in the engine, predicted in analytical studies. It was not studied whether the negative effects of the inflow induced matrix temperature oscillations are general or if they follow some of the same trends as the overall matrix temperature oscillations. The magnitudes of the observed effects of matrix temperature oscillations will most likely be different for engines with different designs.

5. Conclusions

Using a detailed numerical model to study the existing SM5 Stirling engine, we found that the temperature oscillations of the regenerator matrix could be viewed as consisting of two contributions: an overall oscillation of a nearly linear temperature profile and additional inflow induced oscillations near the ends of the regenerator. By mapping the effects of matrix temperature oscil-

lations, we found that, as predicted in the literature, the overall matrix temperature oscillations induced a heat pumping effect and caused a reduction of power output. When the overall matrix temperature oscillations were large, the heat pumping effect could dominate so that the efficiency of the engine was slightly improved. The inflow induced temperature oscillations were found to reduce both the efficiency and the power output.

An optimisation of a new regenerator design where the matrix was divided into three sections was performed for the SM5 engine. The optimisation resulted in the end sections having larger fill factors and wire diameters than the central section. The new design reduced the inflow induced matrix temperature oscillations and intensified the overall matrix temperature oscillations, and it improved the efficiency of the engine from 32.9% to 33.2% while causing a 3% reduction of the power output. By comparison, a maximum electrical efficiency of 33.0% could be achieved with a 2% loss of power using a uniform matrix design with a smaller fill factor and made from thinner wire than the current regenerator of the engine.

A mapping of the performance of the SM5 engine also indicated that an increase in power output could be achieved with a moderate loss of efficiency by choosing a matrix with a larger fill factor and made from thicker wire than the current regenerator.

Acknowledgments

The development of the SM5 Stirling engine is funded by the gas company Naturgas Midt-Nord, the Danish Energy Agency and the EU-Commission. The development of the simulation software used for simulating the SM5 engine has been sponsored by the gas company DONG.

References

[1] Thomas B, Pittman D. Update on the evaluation of different correlations for the flow fiction factor and heat transfer of Stirling engine regenerators. In: 35th Intersociety energy conversion engineering conference, Las Vegas, July 24–28, 2000. p. 76–84.

[2] Gedeon D, Wood JG. Oscillating-flow regenerator test rig: hardware and theory with derived correlations for screens and felts. NASA Contractor Report 198442, February 1996.

[3] Isshiki S, Takasaki Y, Ushiyama I. An experimental study on flow resistance of regenerator wire meshes in oscillating flow. In: 32nd Intersociety energy conversion engineering conference, Honolulu, July 27–August 1, 1997. p. 1027–32.

[4] Siegel A. Experimental investigations on the heat transfer behaviour of wire mesh regenerators in an oscillating flow. In: Proceedings of the European Stirling forum, Osnabrück, February 22–24, 2000. p. 139–47.

[5] Jones JD. A new regenerator theory. In: Proceedings of the 17th Intersociety energy conversion engineering conference, August 8–12, 1982. p. 1656–61.

[6] Jones JD. Performance of a Stirling engine regenerator having finite mass. ASME J Eng Gas Turbines Power 1986;108(October):669–73.

[7] Bauwens L. Near-isothermal regenerator: complete thermal characterization. J Thermophys Heat Transfer 1998; 35(3):414–22.

[8] Gary J, Daney DE, Radebaugh R. A computational model for a regenerator. In: Proceedings of the 3rd cryocooler conference, Boulder, September 17–18, 1984. p. 199–211.

[9] Andersen SK, Carlsen H, Thomsen PG. Simulation of temperature fluctuations in Stirling engine regenerator matrices. In: Proceedings of the 11th international Stirling engine conference, Rome, 2003. p. 120–7.

[10] Carlsen H, Bovin J. Test of 9 kW Stirling engine using biogas as fuel. Proceedings of the 10th international Stirling engine conference. Osnabrück: VDI; 2001.
[11] Andersen SK, Carlsen H, Thomsen PG. Preliminary results from simulation of temperature fluctuations in Stirling engine regenerator matrices. Energy, ECOS2003 Special Issue, in press.
[12] Kühl HD, Schulz S. Modeling of thermal regenerator losses due to axial flow dispersion. In: Proceedings of the 7th international conference on Stirling cycle machines, Tokyo, November 5–8, 1995. p. 1343–8.
[13] Andersen SK, Carlsen H, Thomsen PG. Control volume based modelling in one space dimension of oscillating, compressible flow in reciprocating machines. Simul Modelling Practice and Theory. SIMS2004 Special Issue, in press.
[14] Urieli I, Berchowitz D. Stirling cycle engine analysis. Bristol: Adam Hilger; 1984.
[15] Andersen SK, Carlsen H, Thomsen PG. Parallel shooting methods for finding steady state solutions to engine simulation models. Simul Modeling Practice and Theory, submitted for publication.

(This page intentionally left blank)

Paper C:

Numerical study on transverse asymmetry in the temperature profile of a regenerator in a pulse tube cooler

Submitted on 1/8-2005 for publication in International Journal of Heat and Mass Transfer.

(This page intentionally left blank)

Numerical study on transverse asymmetry in the temperature profile of a regenerator in a pulse tube cooler

Stig Kildegaard Andersen[a], Marc Dietrich[b], Henrik Carlsen[a], Günter Thummes[b]
[a]Dept. of Mech. Eng., Energy Eng. Section, Technical University of Denmark, Kgs. Lyngby, Denmark
[b]Institute of Applied Physics, University of Giessen, D-35392 Giessen, Germany

Abstract
Transverse asymmetry in the temperature profile of the regenerator in a Stirling-type pulse tube cooler as observed in experiments was analysed in a numerical study. The asymmetry was reproduced using a one-dimensional model of the cooler where the regenerator was modelled using two identical parallel regenerator channels. The asymmetry was caused by a circulating flow that was superimposed on the oscillating flow. The primary mechanism driving the circulating flow was due to the wave form of the pressure difference between the ends of the regenerator and the dependence of the instantaneous mass flow rate on the pressure difference and temperature.
Keywords: Pulse Tube Cooler; Regenerator; Temperature profile; Transverse asymmetry; Circulating flow; Streaming

Nomenclature

C_{fd}	Form drag coefficient
C_{sf}	Skin friction coefficient
$\dot{E}_{est}$	Estimated rate of energy transport, W
$\dot{E}_{mat}$	Rate of heat transfer to matrix, W
$\dot{E}_{regloss}$	Regenerator energy flux loss, W
Δp_{fric}	Pressure difference due to friction in the regenerator, Pa
Δp_{reg}	Instantaneous pressure difference between ends of regenerator, Pa
$\widetilde{\Delta p_{reg}}$	Wave form of the time variation of pressure difference between ends of regenerator, Pa
f	Operating frequency, $1/s$
n	Number of wire screens
$\dot{m}$	Mass flow rate, kg/s
p	Pressure, Pa
$\overline{p}$	Space averaged pressure, Pa
Re	Reynolds number
t	Time, s
T	Temperature, K
$\overline{T}$	Space averaged temperature, K
$\overline{V}$	Cup velocity, m/s
w	Open mesh width, m
$\oint$	Integral in time over one cycle of the machine

Greek symbols

ν	Kinematic viscosity, m^2/s
μ	Dynamic viscosity, $kg/(m \cdot s)$
ρ	Density, kg/m^3

1. Introduction

In contrast to the traditional regenerative cryocoolers, such as the Stirling- and Gifford-McMahon (GM)-cryocoolers, the pulse tube cryocooler (PTC) operates without a cold moving

displacer. This feature leads to increased reliability, lower manufacturing costs and reduced mechanical vibrations at the cold head. Stirling-type high frequency PTCs are particularly attractive, since they can be operated with rubbing-free linear compressors that significantly increase the maintenance free operation time of the cooling system (see [1] for a recent review). At present, there is growing interest in large Stirling-type PTCs with electrical input power higher than 4 kW for potential use in gas liquefaction and power applications of superconductors ([2-5]). For hydrodynamic reasons, the up-scaling of the cooler size for high power leads to regenerators with large cross section areas and low aspect ratios. Such a geometry can give rise to unwanted temperature inhomogeneities in the regenerator, as has been recently observed [3-6].

Temperature differences up to 160 K transverse to the main flow direction have been measured in the temperature distribution in the regenerator of a high-power Stirling-type pulse tube cooler targeted for 80 W cooling at 25-30 K [4,5]. In the experiments it appeared that the transverse temperature asymmetry was initiated when the input power to the cooler, and hence the oscillating mass flow through the regenerator, exceeded a temperature dependent critical value. Dependent on the input power and the wire diameter and material of the wire screen mesh in the regenerator the asymmetry would then need on the order of magnitude one hour of cooler operation to evolve to the fully asymmetric temperature distribution. The transverse asymmetry in the temperature profile was shown [4,5] to considerably reduce both the available cooling power and the efficiency of the PTC and to limit the obtainable no-load temperature. It was found that the transverse asymmetry and its detrimental effects on the performance of the PTC could be reduced by increasing the transverse heat conductivity of the regenerator matrix by replacing some of the stainless steel wire screens in the matrix with copper wire screens.

The basic geometry of the problem is illustrated in Fig. 1. A cylindrical regenerator is subjected to the oscillating flow in the PTC. Three Pt-100 temperature sensors were placed at the mid plane of the regenerator interspaced by 120 degrees along the periphery of the regenerator canister. The sensors were expected to show identical temperatures during operation. But at some point in time during the cool down phase of the PTC the temperatures measured by the sensors began to diverge. Once the temperatures measured by the sensors scattered the sensors did not converge back to having the same temperature until the PTC was turned off.

The orientation of the asymmetry was observed to vary between different experimental runs with the same experimental setup, i.e. it varied from experiment to experiment which of the three temperature sensors that measured the highest temperature and which of the temperature sensors that measured the lowest temperature. In some experiments it was even observed that the asymmetry appeared to rotate very slowly with a period of the order of magnitude 5 hours. Due to these observations the transverse asymmetry in the temperature profile was not believed to be caused only by geometric asymmetry in the experimental setup.

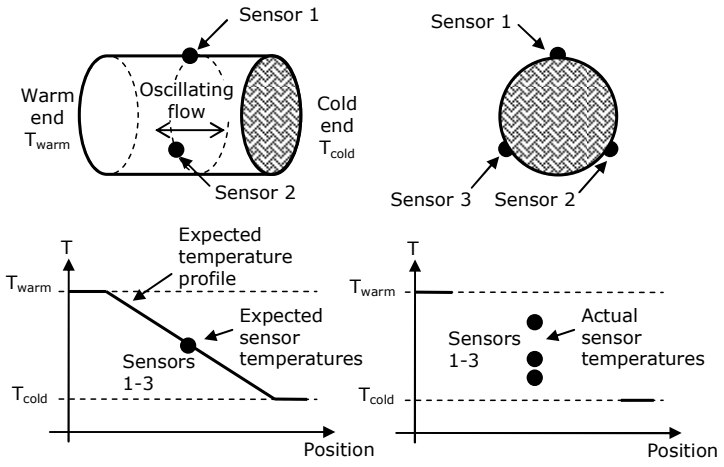


Fig. 1. Basic geometry of problem.

In regenerators there are a number of mechanisms for energy transport transverse to the main flow direction that would be expected to work against transverse asymmetry in a matrix temperature profile. Prime amongst these mechanisms are:

- Conduction in the wires of the regenerator matrix. This conduction will be more powerful in wire screen matrices than in metal felt matrices, because the wires run unbroken across the entire cross section of the regenerator in the wire screens. The magnitude of the energy transport by conduction in the matrix wires will be proportional to the transverse temperature gradient, the thermal conductivity of the matrix material, and to the amount of matrix material in the regenerator. The transverse temperature gradient corresponding to a given temperature difference between opposing sides of the regenerator is inversely proportional to the diameter of the regenerator. The conduction in the wires will hence be most effective in removing transverse temperature differences in regenerators with small diameters. As mentioned above it has been verified experimentally [4,5] that replacing a fraction of the stainless steel wire screens in a regenerator by copper wire screens can reduce the magnitude of a transverse asymmetry in the matrix temperature profile.
- Molecular conduction in the gas in the regenerator. The thermal conductivity of helium is two orders of magnitude smaller than the thermal conductivity of stainless steel matrix wires. For regenerator matrices, which typically have porosities between 50-80 %, the energy transport by conduction in the gas is therefore expected to be much smaller than the conduction in the wires.
- Enhanced transverse energy transport due to the turbulence and mixing in the flow through the porous matrix of the regenerator. Gedeon and Wood [7] have derived a correlation for the axial conduction enhancement due to turbulence in wire screen regenerators. This correlation predicts that the axial energy transport due to turbulence is between one and two orders of magnitude larger than the axial energy transport due to molecular conduction in the gas in the regenerator of the pulse tube cooler from the experimental study described in refs. [4,5]. If the energy transport transverse to the flow direction due to turbulence is of similar magnitude, then this energy transport could be of the same order of magnitude as the conduction in the wires in regenerator with stainless steel wire screens.
- Bulk cross flow induced by the transverse temperature asymmetry. The gas density in a regenerator is almost inversely proportional to the temperature. If a parallel flow with uniform temperature and velocity distributions enters a regenerator matrix where there is transverse asymmetry in the temperature distribution, then part of the mass flow must flow from the hot side towards the cold side inside the regenerator. The magnitude of the energy transport carried by cross flow is difficult to estimate.

Axial energy transport can also smooth out the asymmetry. Imagine, for instance, that the regenerator canister was emptied so that only two identical wires remained in the canister and that these wires were at the same axial position in the regenerator. If bursts of hot and cold gas were alternately sent through the regenerator canister then surely the temperatures of the wires would soon be identical regardless of any differences in their initial temperatures.

The aims of this study were: 1) to reproduce the experimentally observed transverse asymmetry in the matrix temperature profile in numerical simulations, and 2) to identify the mechanisms that can cause and sustain a transverse asymmetry in the temperature profile of a regenerator with no transverse geometric asymmetry. The study was performed in two stages using two separate numerical models.

As the first stage of the study we used a complete simulation model of a PTC, where the regenerator was divided into two parallel regenerator channels, to reproduce the transverse temperature asymmetry in the regenerator. We found that the cause of the asymmetry was a circulating flow in the closed loop formed by the two parallel regenerator channels and the manifold volumes at the ends of the regenerator. This circulating flow was superimposed on the

oscillating flow through the regenerator. The circulating flow (or streaming) amplified any small transverse asymmetry in the regenerator temperatures.

As the second stage we used a separate, simplified model of one regenerator channel to identify the mechanism driving the circulating flow, i.e. to study the influence of the regenerator matrix temperature on the mass flow predicted by the equations for the regenerator. We found that the circulating flow was due to the shape of the pressure difference wave, $\widetilde{\Delta p_{reg}}$, that drives flow through the regenerator, and the dependence of the instantaneous mass flow rate through the regenerator on the instantaneous pressure difference, Δp_{reg}, and the temperature. We also found that the temperature oscillation in the regenerator had a small amplifying effect on the circulating flow and that the oscillation of the pressure in the regenerator damped the circulating flow. The net result was that a regenerator channel would draw in mass from the cold end of the regenerator if the temperature in the channel decreased, and conversely draw mass from the hot end of the regenerator if the temperature increased.

2. Method

In the first stage of the study we used a complete model of a PTC to reproduce the transverse asymmetry in the regenerator temperatures and the circulating flow which caused the transverse asymmetry. In the second stage of the study we used a separate, simple model of one regenerator channel to study the mechanisms that can drive a circulating flow.

2.1 The complete Pulse Tube Cooler Model

The complete PTC model was used to reproduce the experimentally observed transverse regenerator temperature asymmetry in numerical simulations.

The complete PTC model was built using the control volume based approach described by Andersen et al [9,10] for modelling oscillating, compressible flow which is primarily one dimensional. This modelling approach has been successfully validated for both Stirling machines and pulse tube coolers [9,10], and has been used specifically to study regenerators in Stirling engines [11,12]. The model used in this study has been verified by Andersen [10] to produce results in good agreement with the experimental data and with another pulse tube cooler model constructed in the state of the art simulation software *Sage* of Gedeon [8].

The complete PTC model was built so that the regenerator could either be modelled as a single regenerator channel or be divided into two parallel regenerator channels, each with half the cross sectional area of the single regenerator channel. The components of the PTC included in the computational domain are shown in Fig. 2 for the case where the regenerator is divided into two parallel channels.

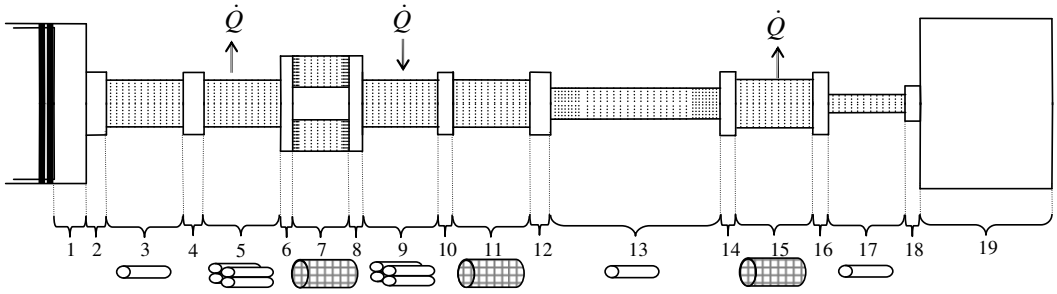
When two regenerator channels were used they were not connected in the transverse direction. Hence the two channels functioned as two identical, parallel, and completely separate regenerators that shared the same boundary conditions. Because the two channels were completely separate the mechanisms for transverse energy transport that normally work against transverse temperature asymmetry were not included in the model. The model thus represents an extreme case: It represents the situation which we would expect to be least stable.

The model with two parallel regenerator channels was verified to give results for symmetric solutions that were identical to the results obtained with one regenerator channel.

In the regenerator of the complete PTC model the flow friction was calculated using the empirical correlation described by Thomas & Pittman [13] with coefficients for data by Gedeon & Wood [7] for flow through wire screen matrices:

$$\Delta p_{fric} = \left(C_{fd} + \frac{C_{sf}}{\mathrm{Re}} \right) \cdot n \cdot \frac{1}{2} \cdot \rho \cdot \overline{V}^2 \cdot \frac{\overline{V}}{\left| \overline{V} \right|},$$
$$\mathrm{Re} = \frac{w \cdot \overline{V}}{\nu}, \quad C_{fd} = 0.5274, \quad \mathrm{C_{sf}} = 68.556 \tag{1}$$

The regenerator had a porosity of 64.4 % and consisted of stainless steel wire screens with a wire diameter of 30 μm, corresponding to an open mesh width of $w = 41.7\mu m$.



Components in computational domain:

1:	Compressor internal volume	11:	Woven mesh flow straightener
2,4,6,...,18:	Manifold volumes	13:	Pulse tube
3:	Transfer line	15:	Hot heat exchanger (Woven mesh type)
5:	After cooler	17:	Inertance tube
7:	Regenerator channels	19:	Reservoir
9:	Cold head		

Fig. 2. Computational domain of pulse tube cooler model.

2.2 Reproducing the transverse asymmetry

A simulation using the complete PTC model with two parallel regenerator channels was started as an initial value problem with a slightly asymmetric initial temperature distribution to see if the asymmetry would increase or decrease with time. The initial asymmetry was introduced by modifying a symmetric solution by making a notch in the axial matrix temperature profile in one of the parallel channels. By performing the experiment twice with the notch in the temperature profile placed in different channels it was also verified that the complete PTC model itself was symmetric.

In the complete PTC model the two regenerator channels and the manifold volumes at the ends of the regenerator formed a closed loop flow path. Gedeon [14] has previously shown that circulating flows are to be expected in oscillating flow in machines with closed loop flow paths that are not geometrically symmetrical. In the complete PTC model, however, the closed loop flow path formed by the two regenerator channels had perfect geometrical symmetry. The only asymmetry that could drive a circulating flow in the numerical experiments was hence the transverse asymmetry in the temperature profiles of the two regenerator channels.

A circulating flow was found and it was tested if the circulating flow could be the main energy transport mechanism increasing the asymmetry. This was tested by comparing the energy transport by the circulating flow to the rates of change in the amounts of energy stored in the matrices of the regenerator channels. The energy transport by the circulating flow was estimated as the circulating mass flow times the enthalpy change in the gas when it travelled from the inflow end of a regenerator channel to the outflow end.

2.3 The separate, simple model of a regenerator channel

A separate, simple model of a single regenerator channel was used to isolate and study the sensitivities to temperature changes of the individual mechanisms that influence the cycle averaged mass flow rates through a regenerator channel.

In the simple regenerator model the regenerator channel was lumped into a single control volume. The mass flow rate through the regenerator was calculated from the average conditions in this control volume and from the pressure difference between the ends of the regenerator. The mass flow was calculated by setting the pressure drop across the regenerator, Δp_{reg}, equal to Δp_{fric} from Eq. (1) and solving for $\overline{V}$. The simple regenerator model thus assumed quasi-steady flow and ignored effects due to the inertia of the gas. The pressure

difference between the ends of the regenerator and the average pressure in the regenerator were prescribed explicitly, so that they could be either simple analytical functions of time or be generated by use of data exported from the complete PTC model or from the *Sage* model of the PTC.

Because there was only one control mass for the regenerator matrix inside the single control volume for the regenerator it was only the average matrix temperature which was known in the simple regenerator model. The matrix temperature could either be assumed constant or be simulated using an energy balance for a lumped control mass. The average temperature difference between the gas and the regenerator matrix was assumed constant but dependent on the flow direction. The average gas temperature in the matrix, which was needed to evaluate Eq. (1), could then be calculated from the matrix temperature and the temperature difference between the gas and the matrix. The gas temperatures outside the ends of the regenerator were set equal to 300K and 60 K. When matrix temperatures were modelled as dynamic, in order to take into account the effect of matrix temperature oscillations, these temperatures were used as the inlet temperatures to the single control volume of the regenerator. The outflow temperature was calculated as the temperature outside the outlet end +/- the constant temperature difference between gas and matrix depending on the flow direction. The method used for calculating the inlet and outlet temperatures only affected the magnitude of the matrix temperature oscillations because it was the before mentioned average gas temperature in the control volume which was used when evaluating Eq. (1).

The calculations performed with the simple regenerator model hence depended mainly on Eq. (1), the boundary conditions, and the gas viscosity used in the calculations. The temperature dependent gas viscosity only varied during individual cycles when the gas temperature varied, i.e. when the matrix temperatures were non-constant and/or there was a finite temperature difference between the gas and the matrix.

The tendencies observed for the single control volume in the simple regenerator model should be the same as the tendencies of individual control volumes in a more complex model, such as the complete PTC model.

2.4 Identifying mechanisms which can contribute to the circulating flow

The simple regenerator model was used for studying how changes in the regenerator matrix temperature influenced the cycle averaged mass flow rate, $f \cdot \oint \dot{m}$, when there was an oscillating mass flow through the regenerator. In practice all the phenomena in the oscillating flow through the regenerator are coupled. But in order to understand the mechanisms driving and opposing the circulating flow, the oscillating flow was split into simpler phenomena which each influence $f \cdot \oint \dot{m}$ through a regenerator channel.

Firstly, it is clear that the wave form of the time dependent pressure difference between the ends of a regenerator channel, $\widetilde{\Delta p_{reg}}$, influenced $f \cdot \oint \dot{m}$. $\widetilde{\Delta p_{reg}}$ was identical for the parallel regenerator channels. But if the contribution from $\widetilde{\Delta p_{reg}}$ to $f \cdot \oint \dot{m}$ through a regenerator channel depended on temperature, then the contributions from $\widetilde{\Delta p_{reg}}$ to $f \cdot \oint \dot{m}$ would be different for different channels with different temperatures.

Secondly, the oscillation of the absolute pressure, $\overline{p}$, in the regenerator also influenced $f \cdot \oint \dot{m}$. In the studied PTC $\overline{p}$ in the regenerator oscillated with an amplitude that was roughly 12 % of the cyclic mean pressure $f \cdot \oint \overline{p}$. The pressure oscillation had a small phase lead of approximately 20 degrees over the pressure difference oscillation, and hence over the mass flow oscillation. The pressure was thus above average when the flow was towards the cold end of the regenerator and below average when the flow was towards the hot end. In Eq. (1) it can be seen that in the limit where Δp_{fric} approaches zero, the term C_{fd} becomes insignificant. In this case

the dependence of Re on ρ cancels the dependence of Δp_{fric} on ρ, so that the velocity becomes:

$$\overline{V} = \frac{2 \cdot w}{C_{sf} \cdot \mu \cdot n} \cdot \Delta p_{fric} \tag{2}$$

The volumetric flow rate therefore becomes independent of $\overline{p}$ as μ is essentially independent of $\overline{p}$. Since the density of the gas is proportional to the pressure the mass flow rate then also becomes proportional to $\overline{p}$. For the range of Δp_{reg} in the PTC model it remains true that the mass flow rate, as computed by use of Eq. (1), was nearly proportional to $\overline{p}$. Because the oscillation in $\overline{p}$ was almost in phase with the oscillation in Δp_{reg}, and hence with the mass flow rate, the oscillation in $\overline{p}$ increased the cycle averaged mass flow rate towards the cold end of a regenerator channel. Because the pressure oscillation was the same for parallel regenerator channels it is, as before, the derivative with respect to $\overline{T}$ of the contribution to $f \cdot \oint \dot{m}$ which is of interest, because this derivative tells us what happens when the temperatures in two parallel regenerator channels are different.

Thirdly, the oscillation in time of the space averaged temperature, $\overline{T}$, in the regenerator also influenced $f \cdot \oint \dot{m}$ through a regenerator channel. The temperature oscillation, which had an amplitude of approximately 0.7 K in the studied PTC, was partly due to the finite heat capacity of the regenerator matrix, and partly due to imperfect heat transfer between gas and matrix and the oscillation of the flow direction. The combined effect of the matrix temperature oscillations and of the imperfect heat transfer was that $\overline{T}$ was a little higher when gas flowed towards the cold end of the regenerator than when gas flowed towards the hot end.

The temperature oscillation caused a slight increase in μ and a slight decrease in ρ when the flow was towards the cold end of the regenerator compared to when the flow was towards the hot end. The oscillation in temperature thus had the effect of decreasing the cycle averaged mass flow through a regenerator channel towards the cold end of the regenerator.

Again it is the derivative of the contribution to $f \cdot \oint \dot{m}$ due to the oscillation in $\overline{T}$ which is of particular interest with respect to the circulating flow.

2.5 Testing the possible contributions to the circulating flow

The contribution to the cycle averaged mass flow rate due to the shape of the pressure difference wave was first studied by mapping how the instantaneous mass flow rate, $\dot{m}$, through the regenerator depended on instantaneous pressure difference between the ends of the regenerator, Δp_{reg}, in the temperature interval from 50 K to 350 K.

The simple regenerator model was then used to integrate $\dot{m}$ during cycles to find the cycle averaged mass flow rates for different pressure difference waves for matrix temperatures between 50 K and 350 K. In each of these integrated cycles $\overline{p}$ and the matrix temperature were kept constant, and it was assumed that $\overline{T}$ was equal to the constant regenerator matrix temperature. The calculations were performed for ideal gas helium with temperature dependent viscosity at $\overline{p} = 2MPa$. The derivative of $f \cdot \oint \dot{m}$ with respect to $\overline{T}$ was then calculated from the results.

The contribution from the oscillation in $\overline{p}$ to $f \cdot \oint \dot{m}$ was studied by including the oscillation in oscillations in $\overline{p}$ into the simple regenerator model. We then repeated the integrations of the instantaneous mass flow rates through a regenerator channel during cycles to determine the value of $f \cdot \oint \dot{m}$ for different matrix temperatures. The only difference between these

integrations and the integrations for constant $\overline{T}$ and $\overline{p}$ in the regenerator was that $\overline{p}$ oscillated in time. Again the derivative of $f \cdot \oint \dot{m}$ with respect to $\overline{T}$ was calculated from the results.

Finally, the contribution from the oscillation in $\overline{T}$ was studied by also including the matrix temperature oscillations into the simple regenerator model, and calculating $f \cdot \oint \dot{m}$ for different initial values of $\overline{T}$. In these calculations it was assumed, based on observations of the solutions to the complete PTC model, that the constant temperature difference between the gas and the matrix was 0.3 K. The derivative of $f \cdot \oint \dot{m}$ with respect to $\overline{T}$ was also calculated for these results.

The results from the simple regenerator model depend strongly on Eq. (1) and the viscosity of the gas is used in Eq. (1). As the final test the calculations for oscillating $\overline{T}$ and $\overline{p}$ were repeated with a constant viscosity equal to the viscosity at 200 K.

3. Results and Discussion

3.1 Reproducing the circulating flow using the complete Pulse Tube Cooler model

The results from the experiment where a simulation using the complete PTC model was started with a slightly asymmetric initial temperature distribution, i.e. with a hand made notch in the temperature profile in one channel, are shown in Fig. 3.

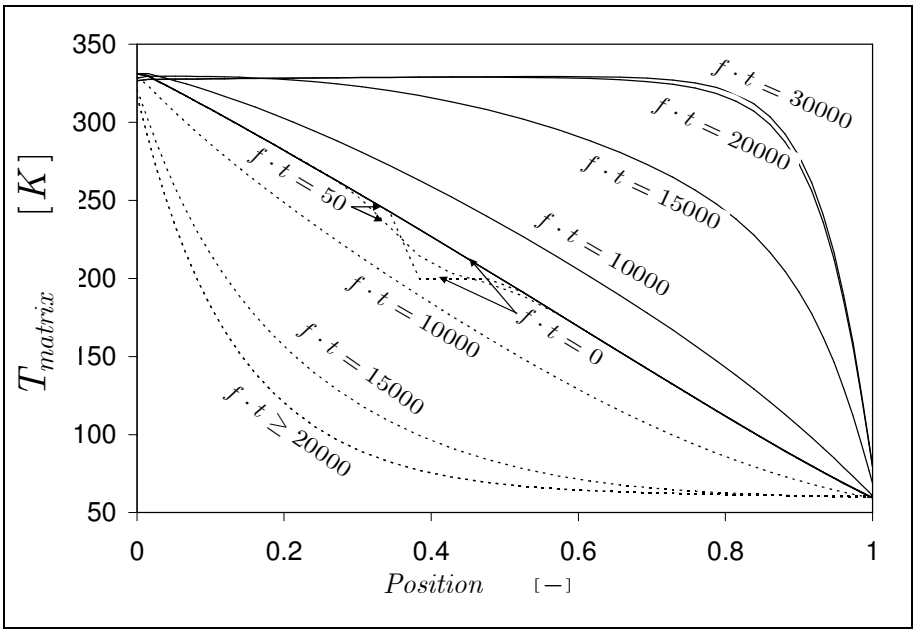


Fig. 3. The evolution in time of the asymmetry in the temperature profiles in the two regenerator channels. At $f \cdot t = 0$ a notch is introduced in the temperature profile of one channel.

Fig. 3 shows that the slightly asymmetric solution was indeed unstable. After a fast initial transient where the notch in the temperature profile was smoothed out by axial energy transport the temperature profiles in the two regenerator channels began to diverge at an accelerating rate. The divergence continued until the temperature profiles were very asymmetric. At the asymmetric periodic steady state solution (number of cycles, $f \cdot t$, > 30000 in Fig. 3) the mechanisms that caused the asymmetry were balanced by axial energy transport mechanisms.

Inspection of the mass flow rates in the two channels in the simulation revealed a circulating flow (streaming) superimposed on the oscillating flow. The mass flow rate of the circulation increased as the asymmetry developed and it increased the total flux loss through the regenerator by up to a factor of 5, as shown in Fig. 4. The increase in the regenerator loss agrees

with the experimental observation that the performance of the PTC suffered when the regenerator matrix temperatures became asymmetric.

The circulating flow drew mass from the cold end of the regenerator into the channel with the lowest average temperature and conversely drew mass from the hot end of the regenerator into the channel with the highest average temperature. The circulating flow removed energy from the coldest channel because the circulating gas was heated when it travelled from the cold end to the hot end of the regenerator. Conversely, the circulating flow transported energy into the warmest regenerator channel, because the circulating gas was cooled on its way from the hot to the cold end. The circulating flow thus amplified the asymmetry in the temperatures of the regenerator channels.

In Fig. 3, the temperature profile in the hottest channel of the regenerator appears to be more extreme than that in the coldest channel. This can be explained by the circulating flow causing approximately the same magnitude of energy transport for both channels while the oscillating mass flow had a smaller amplitude in the hottest channel due to the lower density. Therefore the axial mechanisms that balance the circulating flow must do so with a smaller oscillating mass flow in the hottest channel.

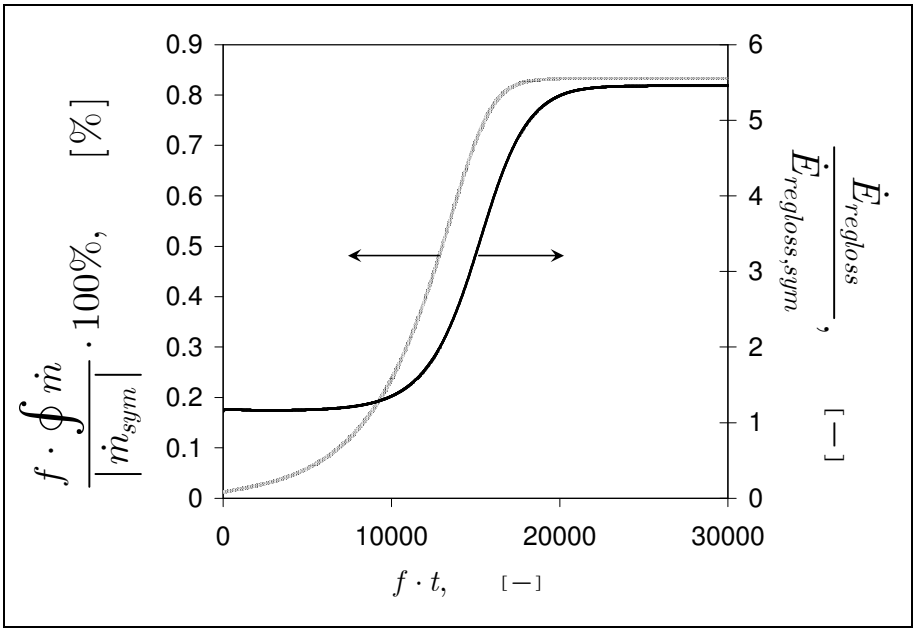


Fig. 4. The evolution in time of the DC flow plotted in percent of the regenerator mass flow amplitude for the symmetric solution, and the evolution of the total energy flux loss through the regenerator, plotted relative to the total energy flux loss for the symmetric solution.

3.2 Energy transport due to circulation and changes in the energy stored in the matrix

The results from test to determine if the circulating flow could be the main energy transport mechanism increasing the asymmetry are shown in Fig. 5. Fig. 5 shows that the rates of change in the energy stored in the matrices in the regenerator channels are smaller than the estimated rates of energy transport due to the circulating flow. The differences between the estimated and the actual rates are moderate in the beginning of the simulation ($f \cdot t < 5000$ in Fig. 5) and then become larger as the asymmetry increases. The differences are largest at the asymmetric periodic steady state solution ($f \cdot t > 30000$ in Fig. 5), where the time averages of the amounts of energy stored in the matrix channels are constant. Fig. 5 shows that the energy transport due to the circulating flow was large enough to be the mechanism that increases the asymmetry.

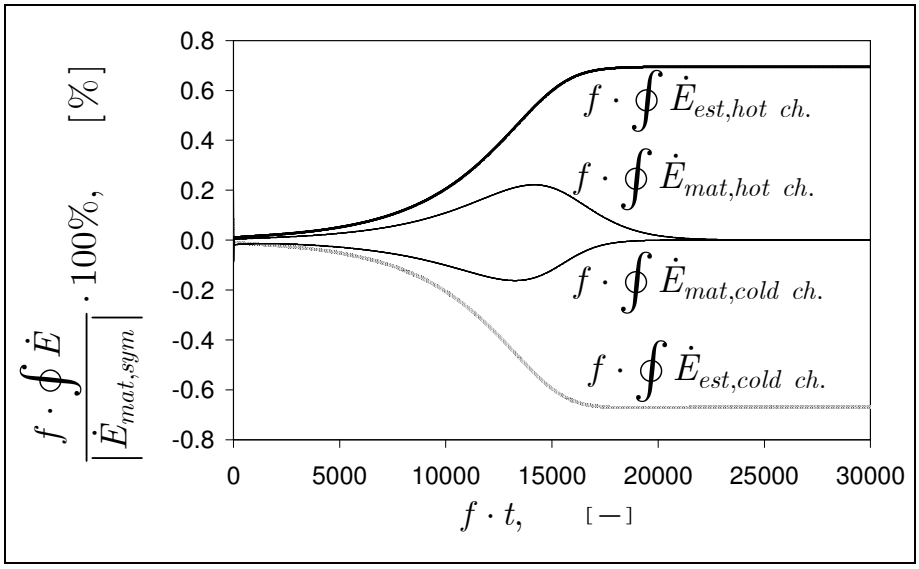


Fig. 5. The time evolutions of the rates of change of energy stored in the matrices of the regenerator channels and of the estimates for the rates of energy transfer by the circulating flow. All quantities have been made dimensionless with the amplitude of the oscillation in the amount of energy stored in the regenerator matrix in the symmetric solution.

3.2 Contribution to the circulating flow from the pressure difference wave form

The results for how the instantaneous mass flow rate in the simple regenerator model depends on the temperature and pressure difference are shown in Fig. 6. Fig. 6 shows that the mass flow rate for a given pressure difference increases with decreasing temperature and that this temperature dependence is largest for small pressure differences.

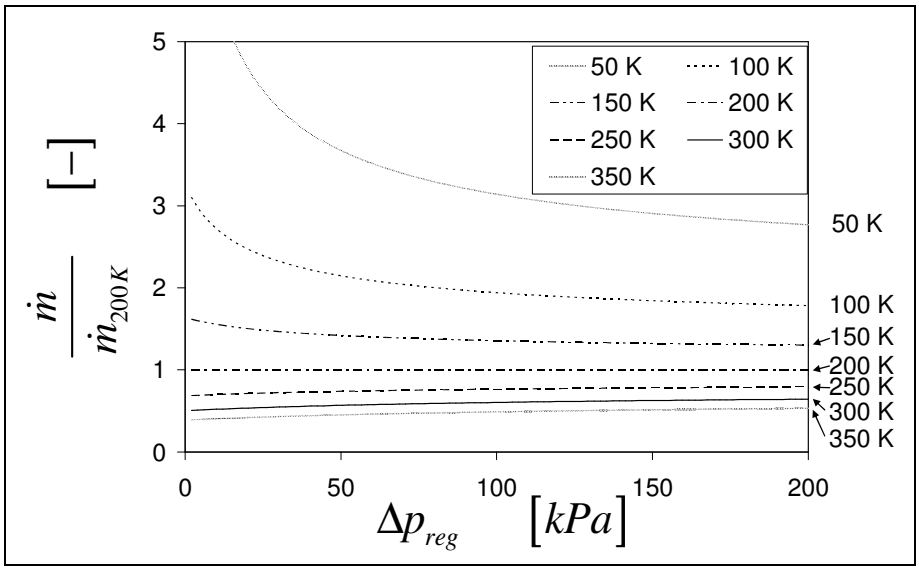


Fig. 6. Regenerator instantaneous mass flow rate relative to regenerator instantaneous mass flow rate at 200 K as function of the pressure difference between the ends of the regenerator.

Fig. 6 shows that the shape of a pressure wave $\widetilde{\Delta p_{reg}}$ that drives flow through two regenerator channels with different $\overline{T}$ can induce a circulating flow. Imagine, as the extreme case, that the pressure difference driving the flow through the regenerator channels looks like the square wave illustrated in Fig. 7. The shape of this pressure wave was calculated so that it gives zero cycle averaged mass flow at 200 K. In Fig. 7 the pressure difference is positive and of magnitude 100 kPa when it drives flow towards the cold end of the regenerator and negative and of magnitude 50 kPa when it drives flow towards the hot end. The pressure difference is negative during 63 % of the cycle. If we apply this square wave to a regenerator channel where $\overline{T}$ is 100 K, then Fig.

6 shows that the flow towards the cold end would be 1.94 times larger (read at $\Delta p_{reg} = 100\ kPa$) and that flow towards the hot end would be 2.15 times larger (read at $\Delta p_{reg} = 50\ kPa$) than for $\overline{T} = 200\ K$. At 100 K there would thus be a net cycle averaged mass flow rate, $f \cdot \oint \dot{m}$, towards the hot end. If we try with $\overline{T}$ larger than 200 K then the result is that there will be a larger cycle averaged mass flow towards the cold end.

Let us define the direction of the mass flow so that it is positive when it is towards the cold end. Then we can say that for the square wave pressure difference wave form from Fig. 7 and a regenerator with constant $\overline{T}$ and a constant $\overline{p}$ of 2 MPa, the derivative of $f \cdot \oint \dot{m}$ with respect to $\overline{T}$ is positive. If we the change sign on the pressure difference wave from Fig. 7, then the largest absolute pressure difference will drive flow towards the hot end. In this case the derivative of $f \cdot \oint \dot{m}$ with respect to $\overline{T}$ will be negative. For a sine wave shaped pressure difference then the derivative of $f \cdot \oint \dot{m}$ with respect to $\overline{T}$ is zero. The derivative of $f \cdot \oint \dot{m}$ with respect to $\overline{T}$ in the regenerator channel thus depends on the shape of the pressure difference wave $\widetilde{\Delta p_{reg}}$.

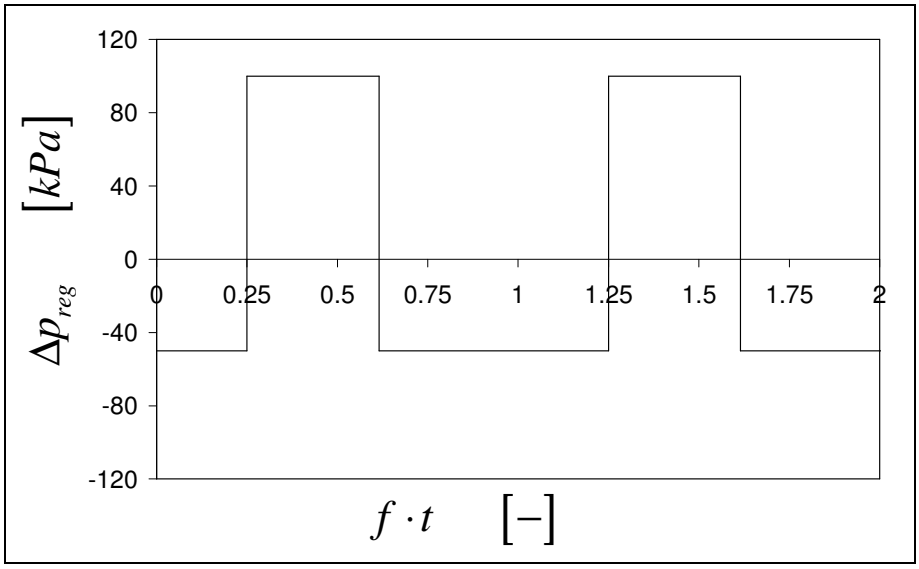


Fig. 7. Square wave pressure difference that gives zero cycle averaged flow in a regenerator channel at 200 K constant temperature and 2 MPa constant space averaged pressure.

Fig. 8 shows the pressure difference wave $\widetilde{\Delta p_{reg}}$ over the regenerator from the complete PTC model plotted so that the pressure difference is positive when it drives flow towards the cold end of the regenerator. The pressure difference wave in Fig. 8 is nearly identical to the pressure difference wave predicted by the *Sage* model of the PTC. The curve in Fig. 8 shares some of the characteristics of the square wave in Fig. 7. The peak values of the positive pressure difference are 10 % larger than the peak values of the negative pressure difference, and the pressure difference is negative during 54 % of the cycle.

Fig. 9 shows the derivative of the net mass flow rate towards the cold end of a regenerator channel with respect to $\overline{T}$ for the pressure difference wave from Fig. 8 when neither $\overline{T}$ or $\overline{p}$ oscillate. The derivative is positive throughout the examined region and largest at low $\overline{T}$. This makes the symmetric situation unstable when there are more than one regenerator channel. Since the derivative of the cycle averaged mass flow rate with respect to $\overline{T}$ is largest at low $\overline{T}$ the situation will become increasingly unstable as the regenerator cools down.

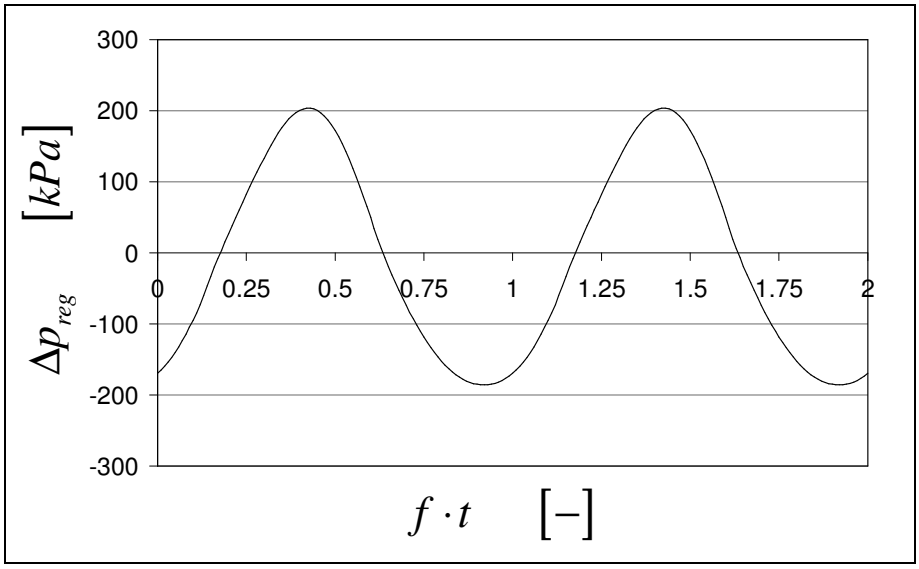


Fig. 8. Pressure difference over the regenerator versus time in the pulse tube cooler model. The pressure difference is positive when the pressure at the warm end is larger than the pressure at the cold end.

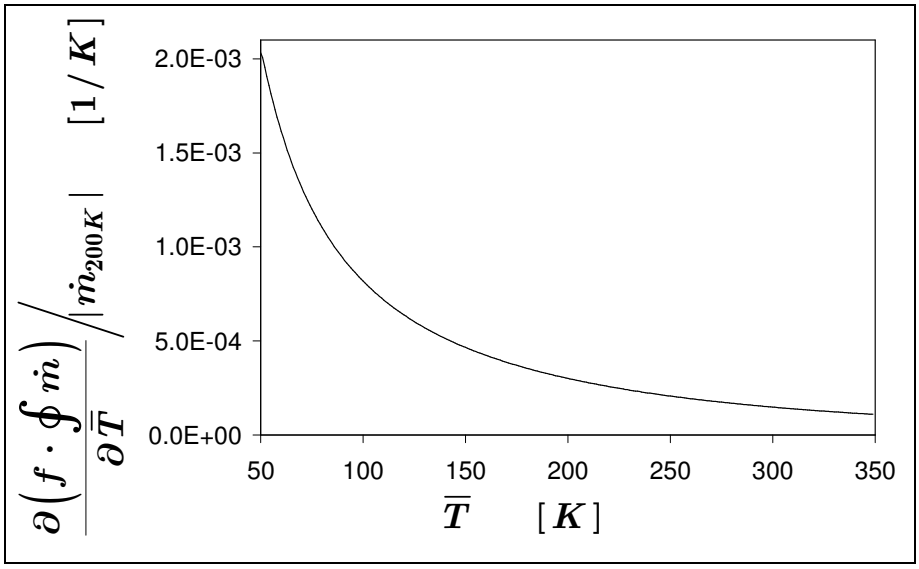


Fig. 9. Derivative of cycle averaged mass flow rate with respect to space averaged temperature $\overline{T}$ calculated at 2 MPa constant space averaged pressure calculated using the simple model of a regenerator. The derivative has been divided by the amplitude of the mass flow oscillation at 200 K in the simple regenerator model.

3.3 Effects of the oscillations in pressure and temperature, and the effect of the temperature dependent viscosity

The results from the test of the contributions to the cycle averaged mass flow rate from the pressure oscillation, temperature oscillation, and the effect of the temperature dependent viscosity are shown in Fig. 10.

At an average regenerator temperature of 200 K the 12 % pressure oscillation increased $f \cdot \oint \dot{m}$ towards the cold end of a regenerator channel by approximately 4.5 % of the amplitude of the mass flow oscillation.

Fig. 10 shows, that the pressure oscillation decreased the derivative with respect to $\overline{T}$ of $f \cdot \oint \dot{m}$ towards the cold end of a regenerator channel. The contribution to $f \cdot \oint \dot{m}$ from the oscillation in $\overline{p}$ hence had a stabilising effect when there was transverse asymmetry in the

temperatures of the regenerator channels, and it did not contribute to the instability of the temperature profiles. Fig. 10 shows that the derivative with respect to $\overline{T}$ of $f \cdot \oint \dot{m}$ towards the cold end of a regenerator channel was reduced by 40 to 75 %, but that is remained positive. The stabilising effect of the oscillation in $\overline{p}$ was therefore not large enough to remove the instability.

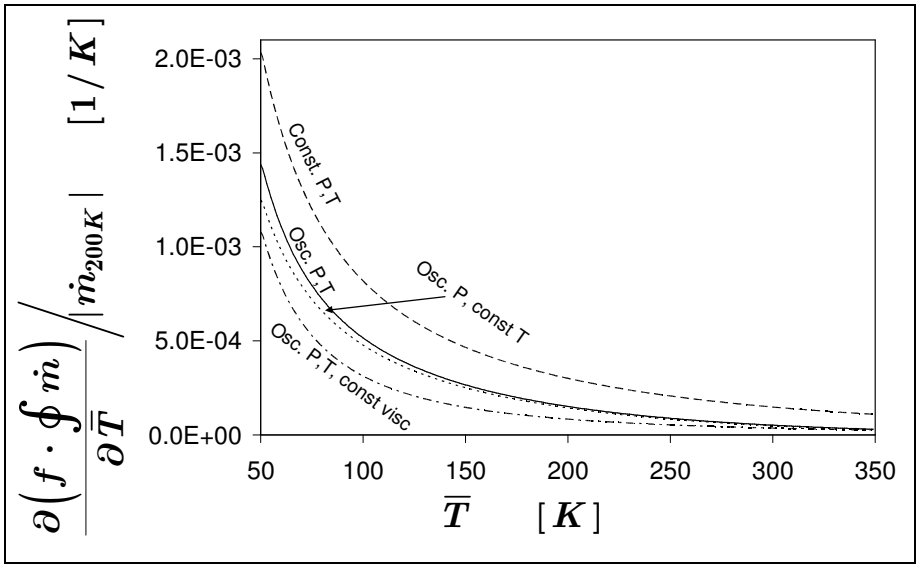


Fig. 10. Derivatives of cycle averaged mass flow rate with respect to space averaged temperature for different combinations of oscillating and constant temperatures and pressures, and constant or temperature dependent viscosity. The derivatives have been divided by the amplitude of the mass flow oscillation at 200 K in the simple regenerator model.

At an average regenerator temperature of 200 K the 1.4 K gas temperature oscillation decreased the cycle averaged mass flow through a regenerator channel towards the cold end of the regenerator by a mere 0.1 % of the amplitude of the mass flow oscillation.

Fig. 10 shows that temperature oscillation slightly increased the derivative with respect to $\overline{T}$ of $f \cdot \oint \dot{m}$ towards the cold end of a regenerator channel. That is because the magnitudes of the derivatives of both μ and ρ with respect to temperature decrease with increasing temperature, and hence the effect due to the oscillation in $\overline{T}$ also decreases with increasing temperature. The temperature oscillation thus contributes to the instability, but the contribution is small.

Fig. 10 shows that removing the temperature dependence of the viscosity reduced the magnitude of the derivative with respect to $\overline{T}$ of $f \cdot \oint \dot{m}$ towards the cold end of a regenerator channel, but that the derivative remained positive. This shows that the temperature dependence of the viscosity contributed to the instability but that even with constant viscosity the flow resistance predicted by Eq. (1) would still lead to instability.

4. Generalisation of the results

In the complete PTC model there was a temperature gradient of 3600 K/m in the regenerator. In a short control volume in the regenerator, however, the variation in temperature inside the control volume was small compared to the average temperature in the control volume. For such a control volume the result of the above analysis, that used a space averaged temperature, should be valid, i.e. the control volume should exhibit the same tendencies as the single control volume of the simple regenerator model.

The effects driving the circulating flow were studied using Eq. (1) with coefficients for data by Gedeon & Wood [7] for flow through wire screen matrices to compute the flow friction in the

regenerator. Thomas and Pitmann [13] provide 8 sets of coefficients for Eq. (1) corresponding to experimental data and to correlations derived from experimental data from different authors for both wire screen and metal felt regenerator matrices. The curve in Fig. 10 for oscillating $\overline{T}$ and $\overline{p}$ was recalculated for each of the 8 sets of coefficients provided by Thomas and Pitmann and there was only a slight spread between the curves for the different sets of coefficients. Hence the results of the above analysis should be valid for both felt and wire screen regenerators.

In this study the regenerator channels in the PTC model were identical and it was necessary to introduce the initial transverse asymmetry in the regenerator temperatures by hand (the notch in one of the channels). In real life it is impossible to create a perfectly symmetrical regenerator or PTC and hence some transverse asymmetry in the regenerator temperatures are to be expected. This transverse temperature asymmetry due to the geometrical asymmetry in a real PTC will hence always be available as a trigger for the temperature dependent effects discussed above.

5. Conclusions

A circulating flow that amplifies transverse asymmetry in the temperature profile in the regenerator of a pulse tube cooler has been reproduced using a complete simulation model of a pulse tube cooler where the regenerator is divided into two identical, parallel regenerator channels. The mechanisms governing the circulating flow have been identified and studied using a separate, simple model of a regenerator.

The primary mechanism driving the circulating flow was found to be due to the shape of the pressure difference wave and the dependence of the instantaneous mass flow rate on the instantaneous pressure difference between the ends of the regenerator and on the temperature in the regenerator. A small contribution to the circulating flow was also linked to the temperature oscillations in the regenerator. These mechanisms caused a regenerator channel to draw in mass from the cold end of the regenerator if the temperature in the channel decreased, and conversely to draw in more mass from the hot end of the regenerator if the temperature in the cannel increased. These mechanisms hence induced a circulating flow that amplified transverse asymmetry in the regenerator matrix temperature profiles. A mechanism due to the oscillations in pressure was found to have the opposite effect and hence worked against transverse asymmetry.

The asymmetry and the circulating flow increased the energy flux loss through the regenerator towards the cold heat exchanger by up to a factor of 5. The circulating flow was thus detrimental to the cooling power and the efficiency of a PTC.

It appears possible to reduce the magnitude of the asymmetry and hence of its detrimental effects on PTC performance by increasing transverse energy transport relative to axial energy transport in the regenerator. This can be done either by increasing the transverse heat conductivity of the regenerator matrix or by putting less heat load on the regenerator, so that the existing transverse heat conductivity is sufficient to keep the asymmetry at an acceptable level. It appears that dividing a single regenerator into two or more parallel regenerators with smaller cross sections is likely to maximise transverse asymmetry because it will inhibit the transverse energy transport that takes place in a single regenerator. It should thus lead to the largest circulating flow and the largest regenerator energy flux loss. Strong cooler losses have been experimentally observed by Kirkconnell in a specially designed small-size PTC with three parallel regenerator tubes [15].

Finally, it should be noted, that the shape of the pressure difference wave depends both on the design and the operating conditions of the PTC. It may be possible to sufficiently modify the shape of the pressure difference wave by changing the design or the operating conditions of the PTC in such a way that the tendency to induce circulating flow and instability is reduced.

Acknowledgements

Computer hardware used for the development of the *MusSim* software was sponsored by the Danish energy company DONG.

References

1. R. Radebaugh, Development of the pulse tube refrigerator as an efficient and reliable cryocooler, Proceedings of the Institute of Refrigeration, 96, London, 1999-00, pp. 11-31.
2. J.H. Zia, A commercial pulse tube cryocooler with 200 W refrigeration at 80 K, Cryocoolers 13, Springer, New York, 2005, pp. 165-171
3. J. Yuan, J. Maguire, Development of a single stage pulse tube refrigerator with linear compressor, Cryocoolers 13, Springer, New York, 2005, pp. 157-163
4. B. Gromoll, N. Huber, M. Dietrich, L.W. Yang, G.Thummes, Development Of A 25 K Pulse Tube Refrigerator For Future HTS-Series Products In Power Engineering, presented at 2005 Cryogenic Engineering Conference, Keystone, Colorado, paper no. C1-S-01, to appear in Advances in Cryogenics Engineering, Volume 51, 2006
5. M. Dietrich, L.W. Yang, G. Thummes, High-power Stirling-type pulse tube cooler: Observation and Elimination of Regenerator Temperature-Inhomogeneities, submitted to Cryogenics
6. G.W. Swift, Thermoacoustics, Acoustical Society of America, 2002, chapter 7.4.4
7. D. Gedeon, J.G. Wood, Oscillating-Flow Regenerator Test Rig: Hardware and Theory With Derived Correlations for Screens and Felts. NASA Contractor Report 198442, 1996
8. D. Gedeon, Sage: Object Oriented Software for Stirling Machine Design, In: Proc. of the 29th Intersociety Energy Conversion and Engineering Conference Vol. 4, American Institute for Aeronautics and Astronautics, Monterey CA, 1994, pp. 1902-1907
9. S.K. Andersen, H. Carlsen, P.G. Thomsen, Control Volume Based Modelling in one Space Dimension of Oscillating, Compressible Flow in Reciprocating Machines, Simulation Modelling Practice and Theory, SIMS 2004 Special Issue, In Press
10. S.K. Andersen, Numerical Simulation of Cyclic Thermodynamic Processes, PhD thesis, Dept. of Mech. Eng., Energy Eng. Section, Technical University of Denmark, 2006
11. S.K. Andersen, H. Carlsen, P.G. Thomsen, Preliminary Results from Simulations of Temperature Fluctuations in Stirling Engine Regenerator Matrices, Energy 31 (2006), pp. 1371-1383
12. S.K. Andersen, H. Carlsen, P.G. Thomsen, Numerical study on optimal Stirling engine regenerator matrix designs taking into account the effects of matrix temperature oscillations, Energy Conversion and Management 47 (2006), pp. 894-908
13. B. Thomas, D. Pittman, Update on the Evaluation of Different Correlations for the Flow Friction Factor and Heat Transfer of Stirling Engine Regenerators, 35th Intersociety Energy Conversion Engineering Conference, Las Vegas, July 24-28 2000, pp. 76-84
14. D. Gedeon, DC gas flows in Stirling and pulse tube cryocoolers, Cryocoolers 9, Plenum Press, New York, 1997, pp. 385-392.
    C.S. Kirkconnell, Experimental investigation of a unique pulse tube expander design, Cryocoolers 10, Kluwer Academic/Plenum Publishers, New York, 1999, pp. 239-247

(This page intentionally left blank)

Paper D:

Numerical study on the appendix gap losses in a Stirling engine

Andersen, S K. Carlsen, H. Thomsen, P G. Numerical study on the appendix gap losses in a Stirling engine. In: Proceedings of the 12th International Stirling Engine Conference and Technology Exhibition, Durham UK, 7-9 September, 2005 ISBN: 0-9535-5582-8, pp. 336-347.

(This page intentionally left blank)

PRELIMINARY RESULTS FROM A NUMERICAL STUDY ON THE APPENDIX GAP LOSSES IN A STIRLING ENGINE

Stig Kildegaard Andersen, Henrik Carlsen
Dept. of Mech Eng., Energy Eng. Section, Technical University of Denmark, Kgs. Lyngby, Denmark
Per Grove Thomsen
Informatics and Mathematical Modelling, Technical University of Denmark, Kgs. Lyngby, Denmark

ABSTRACT

Analytical expressions for the losses in the displacer clearance gap, a.k.a. the appendix gap, have been refined during the last decades. But most real life Stirling engines violate the assumptions behind these expressions and hence the expressions may not be applicable. In this study the gap has been included directly into a one dimensional Stirling engine model. Practical aspects of the method, such as handling the moving wall in the gap while achieving an energy conserving model formulation and handling discontinuous derivatives in the equations, are discussed. A study on the convergence of the spatial discretisation in the gap showed that a relatively coarse discretisation was adequate for studying the appendix gap losses and showed significant variations in the axial wall temperature gradients along the gap. A parameter study on the size of the displacer clearance gap was performed with different algorithms for computing the heat transfer in the gap. The results showed higher losses for small gap sizes but smaller losses for large gap sizes when compared to analytical expressions for the appendix gap losses. The appendix gap losses were found to influence both the heat intake and work output of the engine.

INTRODUCTION

In Stirling engines an elongated displacer piston is often used to separate the compression and expansion volumes. The elongated displacer piston keeps the displacer piston seal remote from the hot working gas temperature in the expansion volume. Between the side of the displacer piston and the cylinder wall there is a clearance gap, often denoted the appendix gap, that extends from the hot end of the displacer piston and down to the displacer piston seal. This clearance gap is filled with the working gas of the engine. The temperature difference between the hot end of the displacer and the location of the seal can be several hundred Kelvin in magnitude, leading to a significant axial temperature gradient in the gap. Due to the pressure oscillation in the engine the pressure also oscillates in the gap and there is an oscillating mass flow in and out of the gap. If the gap is very narrow the gas temperatures along the gap are expected to be governed mainly be heat transfer with the walls in the gap; if the size of the gap is increased then the gas temperatures in the gap will also be influenced by the pressure oscillation in the gap that follows the pressure oscillation in the engine. In the clearance gap heat transfer processes, that constitute losses for the Stirling engine, take place. These loses are known as the appendix gap losses and they are the subject of this study.

In classical Stirling analysis, such as the work of Urieli & Berchowitz [1], these heat transfer processes have been treated as three individual losses that are calculated separately from analytical solutions: a conduction loss, a pumping loss, and a shuttle conduction loss.

- The conduction loss is designed to take into account the heat conduction in the displacer piston and cylinder walls induced by the axial temperature gradient along the gap.
- The pumping loss takes into account the energy transport carried by gas flowing in and out of the clearance gap when the pressure oscillates in the hot expansion cylinder volume.
- The shuttle conduction loss covers the loss caused by the displacer piston moving back and forth along the temperature gradient in the gap. When the displacer piston is close to the top of its stroke it absorbs energy from the cylinder wall, and this heat is released to the cylinder wall again when the displacer piston is close to the bottom of its stroke.

In classical Stirling engine analysis these three losses are applied as independent parasitic loss terms to Stirling simulation results that do not include the losses directly. In a real Stirling engine, however, these losses are mutually coupled with each other and with the thermodynamic cycle of the engine.

Efforts have been put into deriving ever more accurate analytical expressions for the appendix gap losses. Some of this work has been in the context of cryogenics because the appendix gap losses directly reduce the refrigeration power in miniature cryogenic coolers [2]. The analytical solution to a simplified problem, where there is no seal at the bottom of the gap, where temperature gradients are constant and the walls are thin and of uniform thicknesses, has been successively refined to include more effects, such as the finite heat capacity of the walls and even some flow in the appendix gap. A comparison of a number of analytical solutions can be found in the work of Baik and Chang [3]. Unfortunately, most Stirling engines violate several of the assumptions of the analytical solutions for shuttle conduction and hence it is difficult to say how accurate the analytical expressions are for these engines.

In this work we have studied the 9 kW Stirling Engine, SM5, by Carlsen [4]. Some of the general problems related to applying the analytical expressions for the appendix gap also apply to this engine. This is illustrated by Fig. 1 that shows a drawing of the SM5 Stirling engine. In the SM5 engine there is a seal at the bottom of the displacer clearance gap. Also the thickness of the cylinder wall varies significantly along the gap in this engine where the regenerator is placed around the cylinder. At the bottom of the gap near the seal the cylinder wall is thick and the outside of the cylinder wall is cooled by the cooling water in the cooler of the engine. At the middle of the gap the cylinder wall is thin and the outside of the cylinder wall is, presumably, almost adiabatic due to the way the regenerator is packaged and inserted into the engine. Finally, at the top of the gap the cylinder wall is thick and the outside of the wall experiences powerful heat exchange with the working gas in a manifold between the regenerator and the heater tubes. The varying wall thicknesses and the changing boundary conditions on the outside of the cylinder wall make it unlikely that there is a uniform temperature gradient along the gap.

The displacer clearance gap can be included directly into existing one-dimensional Stirling engine models as an alternative to computing the appendix gap losses using the analytically derived expressions. This approach seems favourable since previous attempts to compute the appendix gap losses numerically, such as the work of Huang & Berggren [5], have obtained reasonable agreement with experimental data for actual engines. Also this approach allows the appendix gap losses to be included directly into the governing equations of a Stirling engine model so that the assumption of having no coupling between the losses and the thermodynamic cycle of the engine can be avoided.

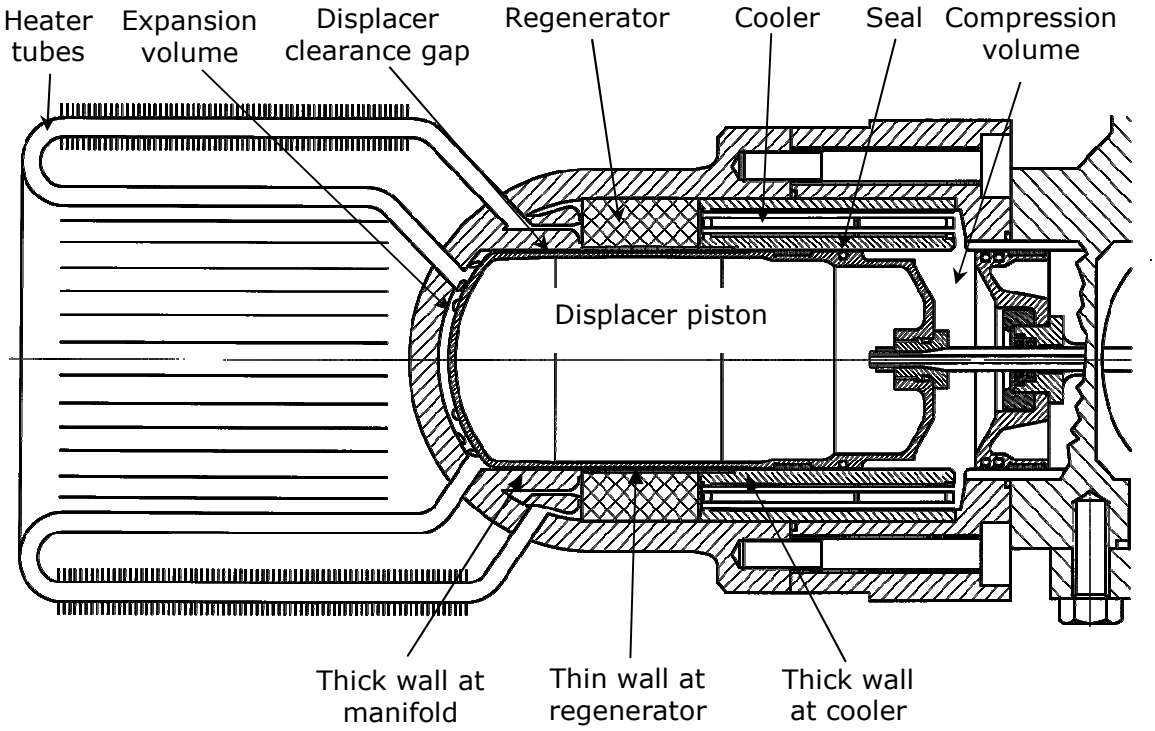


Fig. 1. Drawing of the cylinder region of the 9 kW Stirling engine, SM5.

This paper describes our approach to including the displacer clearance gap in our one dimensional Stirling engine models. The models are built using the control volume based modelling approach which we have previously described in detail [6]. In this modelling approach the gas filled volume of the engine is discretised into two overlapping strings of control volumes using a staggered mesh approach. The governing equations for the gas are derived from the balances for mass, energy, and momentum for the control volumes. The steel of the engine is then discretised into control masses and equations for the temperatures of the control masses are derived from an energy balance for the control masses. The governing equations for the steel and gas are coupled through heat exchange terms in the energy balances for the control masses and control volumes. The main challenges of using this modelling approach to model the displacer clearance gap are computing heat transfer and keeping track of the surfaces that move relative to each other.

Much emphasis was placed on deriving an energy conserving model formulation for the the clearance gap. Energy conservation is important because the overall energy balance for an energy conserving engine model is a valuable test for the correctness of the model and for the accuracy of periodic steady state solutions to the model. The main difficulty in this respect was the motion of the displacer piston relative to the cylinder wall. The focus of this paper is on the practical aspects of making an energy conserving formulation that yielded a system of ordinary differential equations (ODEs) that could be easily solved. Results from a study on the fineness of the discretisation are shown and illustrate that a relatively coarse discretisation was adequate for studying the appendix gap losses. The results also show the varying temperature gradients that were found on the walls of the gap. Results from a parameter study on the clearance gap size where different algorithms were used for computing the heat transfer in the gap are also presented. The results showed higher losses for small gap sizes but smaller losses for large gap sizes when compared to analytical expressions for the appendix gap losses by Urieli & Berchowitz [1]. The appendix gap losses were found to increase the heat intake and decrease the work output of the engine.

METHOD

Discretisation

In order to model the clearance gap it was necessary to use a discretisation that resolved both the gas in the gap, the walls of the displacer piston, and the cylinder wall surrounding the gap. The cylinder wall was discretised using control masses that were fixed in space. The displacer piston wall and the gas in the gap were discretised using control masses and control volumes that followed the motion of the displacer piston.

The discretisation used in the model is illustrated in Fig. 2. The control volumes shown inside the gap in Fig. 2 are the control volumes used for the mass and energy balances in the staggered mesh of the modelling approach [6]. The control volumes used for the mass and energy balances in the gap were directly matched to the control masses on the displacer piston, i.e. each control volume in the gap covered precisely one control mass on the displacer piston wall. The control volumes used for the momentum balances in the staggered mesh flow model are not shown in the figure.

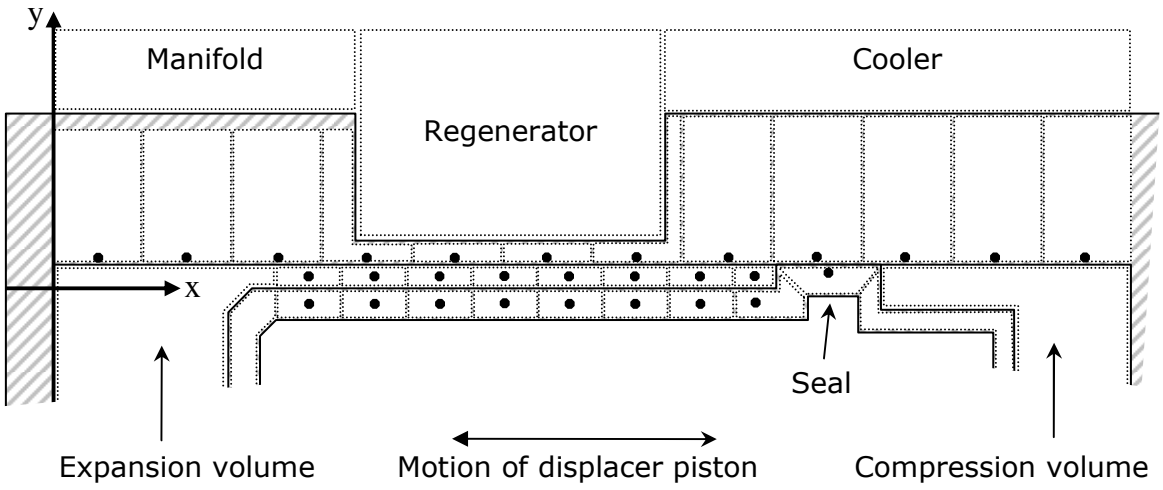


Fig. 2. Discretisation of the displacer wall, cylinder wall, and the displacer clearance gap.

In the discretisation in Fig. 2 there are 4 distinct zones with different heat transfer conditions that need to be considered: The expansion volume, the displacer clearance gap, the seal section, and the compression volume. This means that there are 3 interfaces between the 4 zones. During simulations the positions of these interfaces move relative to the fixed coordinate system shown in Fig. 2 and hence also relative to the cylinder wall.

Wall and gas temperatures

In the model the gas temperatures in the control volumes vary in time. The temperatures of the control masses on the walls of the cylinder and displacer were assumed constant, i.e. it was assumed that the oscillations of the wall surface temperatures were insignificant compared to the oscillation of the gas temperatures; we expect this assumption to become better with increasing gap size. To find the constant wall temperatures the integral conditions were used that the net energy transport to each control mass during a cycle must be zero, i.e. the instantaneous rates of heat transfer including solid conduction to the control mass were integrated over a cycle, and the wall temperatures were found so that the results of these integrations were zero.

Computing and distributing heat transfer

Because the displacer moves back and forth the conditions seen by each of the control masses on the cylinder wall change during the cycle. For instance, the fractions of each of the control masses on the cylinder wall that are exposed to the expansion volume vary with time. In the gap it also varies which of the control volumes in the gap that cover the individual control masses on the cylinder wall.

To deal with this complexity a general scheme using interpolation and integration with splines was devised. The basic idea was to compute the heat transfer pr. unit length at convenient points and to generate a cubic spline through these known points. This spline was then integrated analytically to compute the amounts of heat transfer to individual control masses and control volumes. The use of this scheme will be explained using the displacer clearance gap as example.

In the displacer clearance gap we needed to compute the heat transfer from the cylinder wall to the gas and from the displacer wall to the gas. To compute the heat transfer at a position in the gap we needed, depending on the precise algorithm used for computing the Nusselt number, the conditions in the gas and the wall temperatures at that position. It was a convenient choice to compute the heat transfer at the locations corresponding to the centres of the control volumes in the gap; here the conditions in the gas and the wall temperatures on the displacer were already known and only the wall temperatures on the cylinder wall needed to be interpolated. Because the wall temperatures were assumed constant only one cubic spline through the cylinder wall temperatures needed to be generated for each cycle.

Because the control volumes in the displacer clearance gap were directly matched to the control masses on the displacer wall there was no complication in distributing the heat transfer from the displacer wall to the gas in the gap. To compute and distribute the heat transfer from the cylinder wall to the gap, however, a spline was drawn through the calculated values of heat transfer pr. unit length from the cylinder wall to the gas in the gap. This spline was then integrated analytically over the extents of the individual control masses and control volumes to compute the heat transfer from each of the control masses on the cylinder wall to the gas in the gap, and to compute the heat transfer from the cylinder wall to each of the control volumes in the gap.

Discontinuous derivatives in the governing equations

The motion of the displacer along the cylinder wall brings with it a numerical complication. When the interfaces between the 4 zones with different heat transfer characteristics move over the boundaries between the control masses on the cylinder wall then the rates of heat transfer to the control masses change abruptly, i.e. the derivatives with respect to time of the rates of heat transfer to the control masses become discontinuous in time. The rates of heat transfer to the control masses on the cylinder wall are integrated to find the temperatures on the cylinder wall; the solutions to these equations thus have discontinuities in their second derivatives.

Take, for instance, a control mass on the cylinder wall that sees the expansion volume, and assume that the heat transfer in the gap is more efficient than the heat transfer in the expansion volume. When the displacer moves towards the expansion volume and the interface between the expansion volume and the gap reaches the leftmost edge of the control mass, then the rate of heat transfer to the control mass suddenly begins to change because the gap suddenly begins to cover a fraction of the control mass. When the top of the gap reaches the rightmost edge of the control mass the rate of heat transfer to the control mass suddenly levels at the rate of heat transfer in the gap.

Discontinuous derivatives are a difficulty for many numerical methods used for solving ODEs. Runge-Kutta methods, for instance, approximate solutions to ODEs with smooth and continuous functions between discrete solution points. The smooth functions can only make a good approximation to a solution with a sharp corner if there is a solution point placed exactly at the corner. A time marching Runge-Kutta method that regulates its own step size will, typically, fail some steps due to large error estimates when a sharp corner in a solution is encountered. When the Runge-Kutta method successively reduces its step size as a reaction to the large error estimates, a solution point will eventually land near enough to the corner, so that the first step after the corner does not cause an unacceptable error estimate, and then the solver will march on. Failing steps and changing step sizes, however, wastes computational resources. Also having a solution point near, instead of at, a sharp corner usually causes some inaccuracy in the solution.

We considered two basic options for handling the discontinuous derivatives in the equations for the cylinder wall temperatures: Either the sharp changes in heat transfer at the interfaces between the 4 zones could be smoothed, or the sharp corners could be identified and dealt with numerically. We chose the latter approach because we found no simple answer to how the smoothing at the interfaces should be performed.

In the following we denote it as an *event* when an interface between two heat transfer zones crosses a boundary between two control masses on the cylinder wall. To numerically handle events the model must provide additional information to the numerical solver: The model must provide equations that let the numerical solver determine exactly when events occur, and the model must provide equations for the wall temperature that do not automatically reflect that an event has occurred.

To explain the conditions we used to let the solver determine when events occurred, we denote the positions of the left and right boundaries of a control mass as x_{left} and x_{right} and the position of an interface between two heat transfer zones as $x_{interface}$. If an interface is located on a control mass on the cylinder wall then for that control mass equation (1) will always yield a positive value of g.

$$g = \min(x_{interface} - x_{left}, x_{right} - x_{interface}) \tag{1}$$

If g becomes negative during the integration of the ODEs then it means that the interface has crossed one of the boundaries of the control mass. To locate the point in time when the event occurred the solver must find the point in time where $g = 0$.

In order to avoid generating large error estimates when events are encountered the equations should initially behave as if the events do not occur. Only when the solver has located the exact time of an event should the equations be updated to reflect that the event has occurred.

As an example of how this was done consider the situation where the interface between the expansion volume and the gap is moving towards the expansion volume; specifically, consider the equation with the integral condition for the temperature of the cylinder wall control mass on which the interface is located. The equation for the temperature of this control mass contains a term with an integration of the cubic spline representing the heat transfer pr. unit length from the cylinder wall to the gas in the gap. If we denote the rate of heat transfer pr. unit length from the control mass to the gap as $\dot{Q}'_{cw\_to\_gap}$ then the rate of heat transfer from the control mass to the gap $\dot{Q}_{cm\_to\_gap}$ was computed using Eqs. (2) and (3); Eq. (2) was used until the solver signalled that the event had occurred and Eq. (3) was used after the solver had signalled that the event occurred. Eq. (2) is not physically correct when

$x_{interface} < x_{left}$. But it is valid right up the time of the event and it does not have the discontinuous derivative at the event.

$$\dot{Q}_{cm\_to\_gap} = \int_{x=x_{interface}}^{x=x_{right}} \left(\dot{Q}'_{cw\_to\_gap}\right) dx \tag{2}$$

$$\dot{Q}_{cm\_to\_gap} = \int_{x=x_{left}}^{x=x_{right}} \left(\dot{Q}'_{cw\_to\_gap}\right) dx \tag{3}$$

The big savings in handling the discontinuous derivatives come when using a solver that can cheaply and accurately interpolate inside an already taken step when it solves (1) for the time of an event. We primarily use the semi implicit GERK method by Thomsen [8] that has a continuous extension for interpolation inside already taken steps.

Computing heat transfer in the expansion and compression volumes

In the expansion volume and the compression volume the heat transfer will be governed by the amount of turbulence and the temperatures in these volumes. Directly resolving the turbulence is beyond a one dimensional model so the heat transfer in these volumes must be calculated using empirical correlations. In this work the classical correlation by Woschni [7] has been applied.

Computing heat transfer in the displacer clearance gap

The radial temperature profiles in the gap can be expected to depend strongly on the size of the gap. If the gap is very narrow then the radial temperature profile in the gas will be governed by molecular heat conduction through the gas in the gap and the temperature profile in the gap will be almost linear. If the gap is made larger then it would be expected that compressibility effects, gas flowing in and out of the gap, and eventually even turbulent mixing would govern the temperature profile in the gap so that the heat transfer with the walls in the gap only affects a thin thermal boundary layers near the walls. There may also be a zone with circulation near the seal at the bottom of the gap. Making a one dimensional model that behaves correctly in all these different situations appears impossible.

In this work we ignored effects due to the inlet to the gap and the circulation at the bottom of the gap, and we ignored the influence of any leakage flow across the seal on the radial temperature profiles in the gap. We implemented two methods for computing the instantaneous rates of heat transfer between the walls and the gas in the gap.

In the first method the radial temperature profiles in the gap were approximated with quadratic polynomials. The polynomials were constructed from the conditions that the gas temperatures must be equal to the wall temperatures at the wall, and that the mean temperatures of the temperature profiles must be equal to the mean gas temperature. With this approach the heat fluxes from the walls $\dot{Q}''_{Quadratic}$ were calculated using (4), where T_{cw} is the temperature of the cylinder wall, T_{dw} is the temperature of the displacer wall, T_{gas} and k are the average temperature and the thermal conductivity of the gas, and a is the size of the gap.

$$\dot{Q}''_{Quadratic} = \begin{cases} \frac{k}{a} \cdot \left(4 \cdot T_{dw} + 2 \cdot T_{cw} - 6 \cdot T_{gas}\right), & displacer\ wall \\ \frac{k}{a} \cdot \left(2 \cdot T_{dw} + 4 \cdot T_{cw} - 6 \cdot T_{gas}\right), & cylinder\ wall \end{cases} \tag{4}$$

For very small gaps (4) will produce the expected linear radial temperature profiles. The approach should be better at larger gap sizes than simply assuming a linear temperature profile but for large gaps results will likely not remain accurate.

In the second approach we used the Nusselt number correlations (5) experimentally determined by Huang & Berggren [5] for the P40-R test engine.

$$Nu_{D_h} = \begin{cases} 4 + 0.4456 \cdot \mathrm{Re}_{D_h}{}^{0.322}, & displacer\ wall \\ 4 + 0.7093 \cdot \mathrm{Re}_{D_h}{}^{0.415}, & cylinder\ wall \end{cases}, \quad D_h = 2 \cdot a \tag{5}$$

If the Reynolds number is zero then (5) also reduces to the case where the temperature profile is linear. For working gasses like Helium, however, gaps must be very small if friction is to have a significant impact on the velocities in the gap. We therefore expect (5) to overestimate the heat transfer in small gaps. For large gaps, however, (5) might be more accurate than (4).

The displacer seal section

In the displacer seal section heat transfer was modelled as taking place between two parallel plates separated by a very narrow gap, i.e. with a linear temperature profile in the gas going from the temperature of the displacer at the seal to the temperature of the cylinder wall at the location of the seal. In this study we assumed that no gas flowed through the seal.

Test for convergence of the spatial discretisation

In order to test the spatial discretisation, and to get a feel for the shapes of the wall temperature profiles, a solution to the Stirling engine model was recalculated with different numbers of control masses and control volumes. A solution was first calculated with 15 control masses on the cylinder wall and 10 control volumes in the gap using the quadratic temperature profiles; since the control volumes in the gap and the control masses on the displacer piston wall in the gap were directly matched there was thus also 10 control masses on the displacer piston wall in the gap. The solution was then recalculated twice, each time doubling the number of control masses and control volumes. The discretisations used in the test are summarised in Table 1. The PV work, heat intake, total appendix gap loss, and wall temperature profiles were compared for the three solutions. The total appendix gap loss was computed as the amount of heat conducted directly from the cylinder wall to the cooling water of the cooler.

Table 1. The numbers of control masses and control volumes used when testing the convergence of the spatial discretisation.

	CMs on disp.	CVs in gap	CMs on cyl wall
Case 1	10	10	15
Case 2	20	20	30
Case 3	40	40	60

Comparison of results for varying gap sizes for the different methods

From an engine design point of view it is interesting to know the optimum size of the displacer clearance gap. The gap should be small enough to effectively insulate the displacer seal from the hot gas in the expansion volume and with that constraint it should have the size that results in the best engine performance. Therefore a parameter variation was performed for the size of the displacer gap using (4) and (5) for computing heat transfer in the gap. The gap size was varied between 0.05 mm and 5 mm. The PV work, heat intake, and total appendix gap loss were recorded for the calculated solutions.

In this study we focussed on the effects due to the heat transfer between walls and gas in the gap. To remove effects due to changes in dead volume and in metal cross section areas from the results of the parameter variation, the parameter variation was also performed with a model where the heat fluxes computed using (4) were multiplied by 0.001, i.e. where the walls in the gap were almost adiabatic. Of the three classical appendix gap losses these results hence included mainly the conduction loss. When the walls were almost adiabatic the main influences on performance from increasing the gap size were from the added dead volume and from the reduced steel cross section area available for conduction in the displacer. Estimates for the effects of the pumping loss, shuttle conduction, and the change in the conduction loss from the heat transfer in the gap were calculated by subtracting the results for near adiabatic walls from the results with full heat transfer in the gap. The results were compared to analytical expressions for the appendix gap losses by Urieli & Berchowitz [1]

NUMERICAL RESULTS

Table 2 shows the total appendix gap loss, the heat intake, and the PV-work for the three solutions calculated with the discretisations from Table 1. Fig. 3 shows the wall temperatures on the cylinder wall and on the displacer piston for the corresponding solutions. In Fig. 3 the wall temperatures are plotted against the dimensionless distance from the top of the cylinder wall; the displacer is positioned at its bottom dead center position. The different boundary conditions on the outside of the cylinder wall, that were explained in the introduction, are indicated on the figure.

Table 2. Gap losses, heat intake, and PV work from the three solutions with varying fineness of the spatial discretisation.

	App. Gap. Loss [kW]	Q_dot_in [kW]	PV work [kW]
Case 1: 10 / 15	3.118	32.515	13.276
Case 2: 20 / 30	3.098	32.500	13.276
Case 3: 40 / 60	3.069	32.470	13.277

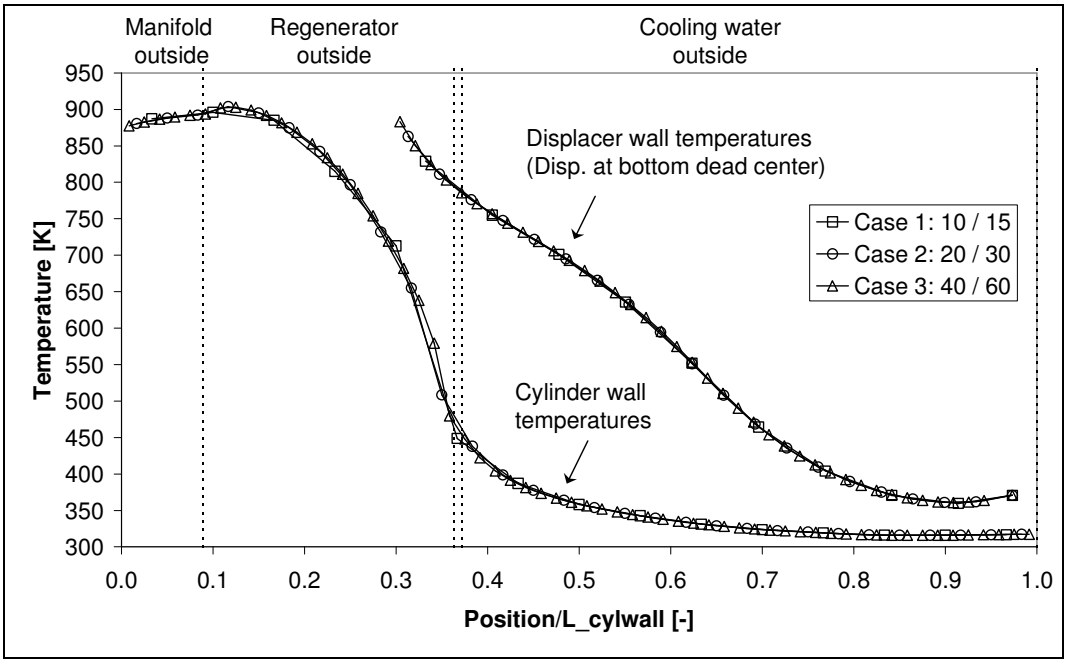


Fig. 3. Displacer and cylinder wall temperatures for varying fineness of the spatial discretisation.

Fig. 4 shows the appendix gap losses calculated using (4) and (5) for gap sizes between 0.05 and 5 mm. The results calculated for a nearly adiabatic gap have been subtracted from the results calculated with heat transfer in the gap to produce the data plotted in Fig. 4. The shuttle loss, pumping loss, and the sum of the shuttle and pumping losses calculated using the formulas of Urieli & Berchowitz are also plotted in the figure.

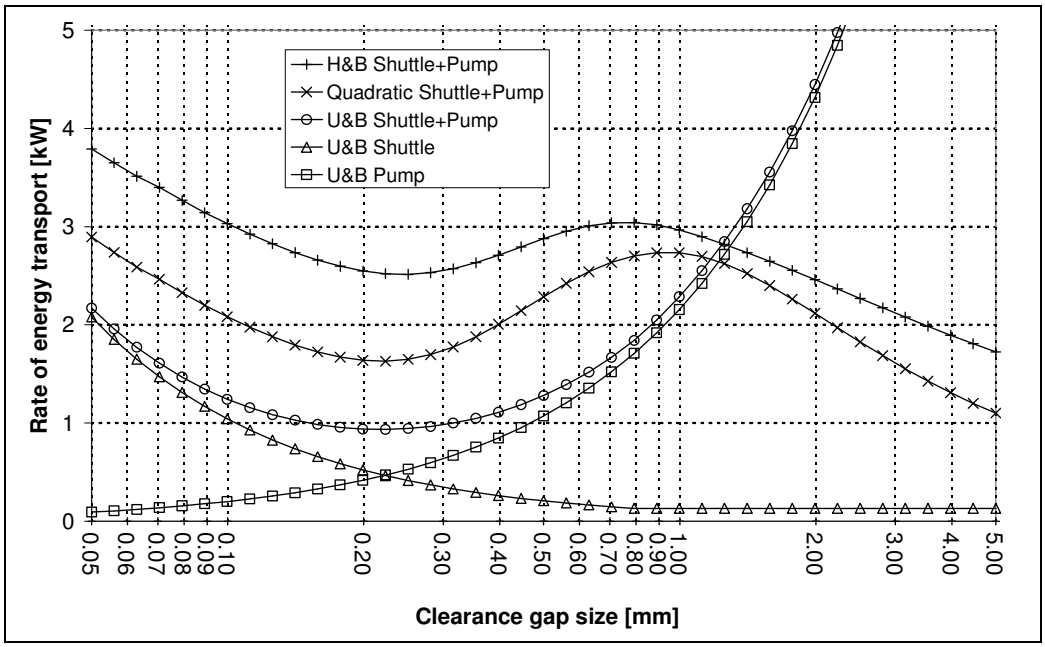


Fig. 4. Gap losses for quadratic temperature profiles, the correlations by Huang and Berggren (H&B), and the shuttle and pumping loss correlations by Urieli and Berchowitz (U&B)**.**

Fig. 5 shows the differences in heat intake, PV work, and appendix gap losses caused by the heat transfer in the gap calculated using (4) and (5) for gap sizes between 0.05 and 5 mm.

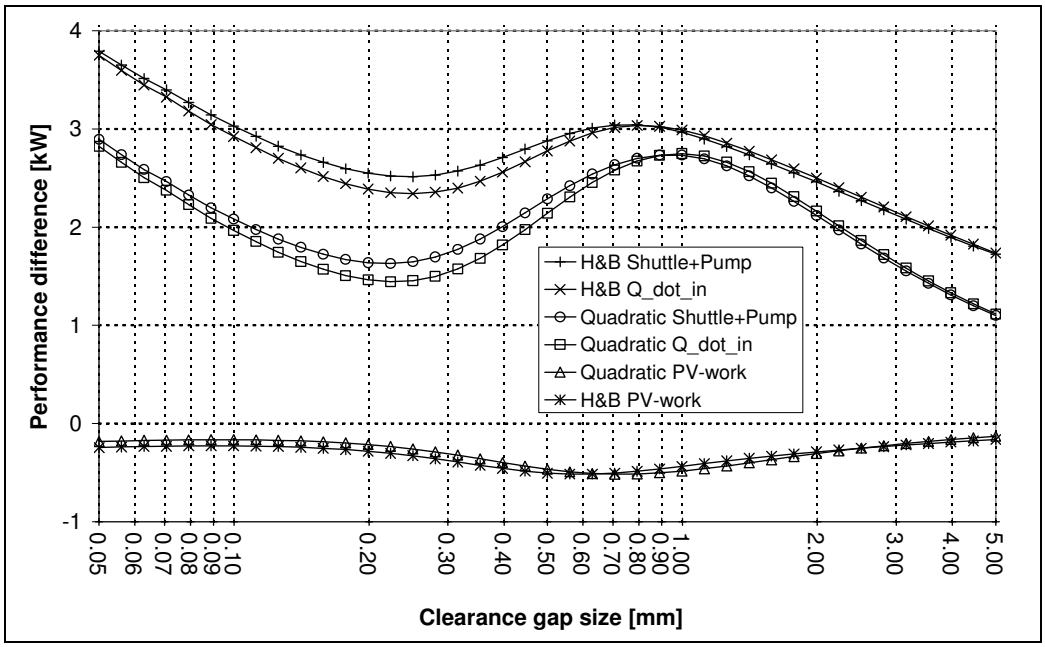


Fig. 5. Impact on engine performance by the shuttle and pumping losses for quadratic temperature profiles and the Nusselt number correlation by Huang and Berggren (H&B).

DISCUSSION

The results in Table 2 show that the calculated appendix gap loss changed less than 1.6 %, or approx. 50 W, when the discretisation was refined. A change of less than 50 W was also seen in the heat intake. The PV work was practically unaffected by the discretisation.

The results in Fig. 3 show that the wall temperature profile on the displacer was practically independent of the discretisation. Changing the fineness of the discretisation caused slight changes in the cylinder wall temperature profile at the positions of the regenerator but the temperature profile on the cylinder wall did appear to be converging.

Based on these results it was decided that it was adequate to use the coarsest of the tested discretisations, i.e. 10 control masses on the displacer in the gap and 15 control masses of the cylinder wall, for the remainder of the study.

The wall temperatures plotted in Fig. 3 show significant variations in the axial temperature gradient, especially on the cylinder wall. The axial temperature gradients are small at the top manifold volume where the cylinder wall is thick, and at the cooler where the wall is also thick and where the outside of the cylinder wall is subjected to efficient cooling by the cooling water from the cooler. The temperature gradients are steep at the thin section of the cylinder wall surrounded by the regenerator.

Fig. 4 shows that for gap sizes smaller than 0.5 mm the methods all give the same characteristic curve shape with minima at a gap size of approximately 0.23 mm. The increase in the appendix gap losses at gap sizes below 0.23 mm is caused by the increase in the shuttle conduction with smaller gap size. The increase in the appendix gap losses for gap sizes above 0.23 mm is due to more energy being carried into the gap from the expansion volume by the oscillating mass flow in the gap. In this region the losses predicted using the quadratic temperature profiles are between 0.7 and 0.9 kW larger than the losses predicted by the formula of Urieli & Berchowitz and the losses predicted using the correlations by Huang and Berggren are between 1.6 and 1.8 kW larger than the losses predicted by the formula of Urieli & Berchowitz.

In the formula by Urieli and Berchowitz the temperature gradient is assumed to be uniform along the gap and this results in a smaller temperature gradient than in our results at the position where the regenerator surrounds the cylinder wall. Part of the difference between the losses we calculate with (4) and (5) and the losses predicted by the formula of Urieli & Berchowitz were likely due to the differences in the temperature gradients. The differences between the losses calculated with (4) and (5) for very small gap sizes were attributed to the Reynolds number dependence of (5).

Fig. 4 also shows that the analytical solution by Urieli & Berchowitz gives very different results than what was calculated using (4) and (5) for gap sizes above 0.5 mm. In the analytical solution the pumping loss kept increasing as the size of the gap was increased. In the results calculated using (4) and (5) the appendix gaps losses had maxima at gap sizes of approximately 1 mm and 0.8 mm and the losses decreased when the gap size was further increased. The decrease in the losses for large gap sizes when using (4) and (5) to compute heat transfer in the gap were attributed to the heat transfer between the walls and the gas in the gap becoming less efficient when the hydraulic diameter was increased. In general, it was noted that the tendencies in the results calculated using (4) and (5) were quite similar even though the methods, and their origins, are very different.

Fig. 5 shows that the appendix gap losses calculated using (4) and (5) translated very closely into increases in the heat intake for the engine. The appendix gap losses also reduced the PV power output from the engine; this reduction was largest at the maxima in the gap losses at 0.8 to 1 mm even though the gap losses were larger at lowest end of the studied range of gap sizes.

This was attributed to the way that energy was transported into the gap. At gap sizes of 0.8 to 1 mm, energy was mainly carried into the gap by the gas from the expansion cylinder volume; the heat transfer in the gap hence drained energy directly from the gas when the pressure in the engine was increasing and this reduced the amplitude of the pressure oscillation in the engine. For very small gap sizes, where the shuttle conduction mechanism constituted the bulk of the appendix gap losses, the energy transport into the gap was mainly by shuttle conduction from the thick wall section of the top of the cylinder. This cooled down the top of the cylinder wall so that it absorbed more energy from the gas in the expansion space; but this heat exchange between the cylinder wall and the gas in the expansion space occurred with a different phase angle than the flow of gas into the gap and it had less impact on the pressure oscillation in the engine.

CONCLUSIONS

In this paper we have described our approach for including the displacer clearance gap, a.k.a. the appendix gap, directly in our one dimensional Stirling engine models.

A study on the convergence of the spatial discretisation used in the gap showed that a relatively coarse discretisation was adequate for studying the appendix gap losses. The wall temperature profiles calculated during the study showed that the axial gradients in the wall temperatures varied significantly along the gap.

A parameter study on the size of the displacer clearance gap was performed with different algorithms for computing the wall to gas heat transfer in the gap, and the results were compared to analytical expressions for the appendix gap losses by Urieli & Berchowith. The simulations resulted in higher losses for small gap sizes but smaller losses for large gap sizes compared to the analytical expressions. The impact of the appendix gap losses on the heat intake and PV work output of the engine was also studied. A close correspondence was found between the appendix gap losses and an increased heat intake of the engine. The appendix gap losses were also found to reduce the work output of the engine; especially for relatively large gaps where the pumping loss dominated the appendix gap losses.

ACKNOWLEDMENTS

The development of the SM5 Stirling engine is funded by the gas company Naturgas Midt-Nord, the Danish Energy Agency and the EU-Commission. The development of the simulation software used for simulating the SM5 engine has been sponsored by the gas company DONG

REFERENCES

[1] Urieli, Israel. Berchowitz, David M. Stirling Cycle Engine Analysis. Adam Hilger Ltd. 1984

[2] Chang, Ho-Myung. Park, Dae-Jong. Jeong, Sangkwon. Effect of gap flow on shuttle heat transfer. Cryogenics 40 (2000). 159-166. 2000.

[3] Baik, Jong Hoon. Chang, Ho-Myung. An exact solution for shuttle heat transfer. Cryogenics (35) 1995. 9-13. 2000.

[4] Carlsen, Henrik. Bovin, Jonas. Test of 9 kWStirling engine using biogas as fuel. 10th International Stirling Engine Conference. VDI. Osnabrück. September 2001

[5] Huang, S.C., Berggren, Robert. Evaluation of Stirling Engine appendix gap losses. 21st Intersociety Energy Conversion Engineering Conference: Advancing Toward Technology Breakout in Energy Conversion. 562-568. 1986.
[6] Andersen, S.K. Carlsen, H. Thomsen, P.G. Control Volume Based Modelling of Compressible Flow in Reciprocating Machines. In: B. Elmegaard, J. Sporring, K. Erleben, K. Sørensen (eds.) Proc. SIMS 2004. Technical University of Denmark. Copenhagen. 2004. pp. 119-128
[7] Woschni, G. Universally Applicable Equation for the Instantaneous Heat Transfer Coefficient in the Internal Combustion Engine. SAE Paper 670931. 1967.
[8] Thomsen, P.G. A generalized Runge Kutta Method of Order Three. Technical Report IMM-REP-2002-07. Department of Mathematical Modelling. DTU Denmark. 2002.